DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

DETECTION OF DATA INJECTION ATTACKS FOR SMART GRID SECURITY

by
Alirıza YAVUZ

March, 2014
İZMİR

DETECTION OF DATA INJECTION ATTACKS FOR SMART GRID SECURITY

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Degree of Master of Science
in Electrical and Electronics Engineering

by

Alirıza YAVUZ

March, 2014
İZMİR

M.Sc THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **"DETECTION OF DATA INJECTION ATTACKS FOR SMART GRID SECURITY"** completed by **ALİRIZA YAVUZ** under supervision of **ASST. PROF. DR. M. EMRE ÇEK** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. M. Emre ÇEK

Supervisor

Doç. Dr. Olcay AKAY

(Jury Member)

Yrd. Doç. Dr. Savaş ŞAHİN

(Jury Member)

Prof. Dr. Ayşe OKUR

Director

Graduate School of Natural and Applied Sciences

ACKNOWLEDGEMENTS

I would like to express my respect and thanks to my supervisor Asst. Prof. Dr. M. Emre ÇEK for his supporting guidance, precious advices, and encouragement throughout this thesis studies. I am also indebted to him for his serious contributions, attention and motivation to complete the thesis.

I would like to thank Assoc. Prof. Dr. Olcay AKAY for his precious contributions, key ideas and advices for completing this study.

I wish also like to thank my family for supporting me throughout my life.

Alirıza YAVUZ

DETECTION OF DATA INJECTION ATTACKS FOR SMART GRID SECURITY

ABSTRACT

In this thesis, static state estimation in smart grid is investigated for non-Gaussian environments. The noise model in state estimation is widely assumed to be Gaussian distributions. But in real world applications, the process noise is impulsive in nature. Alpha-stable distributions are proposed and implemented for constructing impulsive noise. Parameters of alpha-stable distributions are introduced and their impact on noise is discussed. Furthermore, robust m-filters like median, meridian and myriad are presented and compared to weighted least squares (WLS) which is traditionally used for state estimation. MATLAB simulations are performed for comparing performance of filters in impulsive noisy environment.

The importance of attack detection is emphasized for security of smart grid and data injection attack model is defined as a DC offset on measurements. The desired idea is detecting attack as quickly as possible which is named quickest detection problem. In quickest detection, there is a tradeoff between detection speed and detection reliability. CUSUM algorithm which is a sequential analysis for detecting change is proposed and implemented for two-sided tabular form of CUSUM combined with statistical hypothesis tests. Threshold value for CUSUM determines the performance of attack detection. Impact of threshold value selection on detection ratio, false alarm ratio and average run length is investigated for different alpha values. Simulations are performed in MATLAB environment and results are discussed in detail.

Keywords: Smart grid security, state estimation for non-Gaussian environments, robust filters, data injection attack detection, quickest detection.

AKILLI ŞEBEKE GÜVENLİĞİ İÇİN VERİ ENJEKSİYON SALDIRILARININ KESTİRİMİ

ÖZ

Bu tezde, Gauss olmayan ortamlarda akıllı şebekeler için statik durum kestirimi araştırılmıştır. Durum kestiriminde gürültünün Gauss olduğu yaygın olarak kabul edilir. Fakat gerçek dünya uygulamalarında gürültü aslında dürtüseldir. Alfa-kararlı dağılımlar önerilmiş ve gürültünün oluşturulmasında kullanılmıştır. Alfa-kararlı dağılımın parametrelerine değinilmiş ve bu parametrelerin gürültü üzerindeki etkisi tartışılmıştır. Ayrıca meridian, median ve myriad gibi gürbüz m-filtrelere değinilmiş ve bu filtreler durum kestiriminde geleneksel olarak kullanılan ağırlıklı en küçük kareler (WLS) yöntemi ile kıyaslanmıştır. Filtrelerin gürültü ortamlardaki performanslarını kıyaslamak için MATLAB simülasyonu gerçekleştirilmiştir.

Akıllı şebekenin güvenliği için saldırı kestiriminin önemi vurgulanmış ve veri injeksiyon saldırısı, ölçümler üzerindeki DC kayma olarak tanımlanmıştır. Uygulanmak istenilen düşünce, saldırının olabildiğince hızlı tespit edilmesidir. Bu durum hızlı tespit problemi olarak adlandırılır. Hızlı tespit kullanımında, tespit hızı ile tespit güvenilirliği arasında bir tercih söz konusudur. Değişimin tespiti için sıralı bir analiz olan CUSUM algoritması önerilmiş ve iki yönlü çizelge formundaki CUSUM ile istatistiksel hipotez testi birleştirilerek uygulanmıştır. CUSUM için eşik değeri saldırı tespitinin performansını belirlemektedir. Eşik değeri seçiminin tespit oranı, yanlış alarm oranı ve ortalama çalışma süresi üzerindeki etkisi farklı alfa değerleri için araştırılmıştır. Simülasyonlar MATLAB ortamında gerçekleştirilmiş ve sonuçlar detaylı olarak tartışılmıştır.

Anahtar Kelimeler: Akıllı şebeke güvenliği, akıllı şebekelerde Gauss olmayan ortamlarda durum kestirimi, gürbüz filtreler, veri injeksiyon saldırılarının tespiti, hızlı tespit.

CONTENTS

Page

M.Sc THESIS EXAMINATION RESULT FORM ii
ACKNOWLEDGEMENTS iii
ABSTRACT iv
ÖZ v
LIST OF FIGURES ix
LIST OF TABLES xi

CHAPTER ONE - INTRODUCTION 1

CHAPTER TWO - STATE ESTIMATION IN POWER GRID 4

2.1 Static State Estimation 5
2.2 Weighted Least Squares (WLS) 7
2.3 Non-Gaussian and Impulsive Noise Approach 7

CHAPTER THREE - ROBUST M-FILTERS FOR STATE ESTIMATON 9

3.1 Robust M-Filters 9
3.2 Sample Median Filter 9
3.3 Sample Meridian Filter 11
3.4 Sample Myriad Filter 12

CHAPTER FOUR - ALPHA STABLE DISTRIBUTIONS 14

4.1 Alpha-Stable Distributions in Real Form 14
4.2 Alpha-Stable Distributions in Complex Form 16

CHAPTER FIVE - SIMULATIONS OF STATE ESTIMATION 18

5.1 Self-Comparison of Filters for Different Alpha Values 19
5.1.1 Performance of Sample Median Filter 19
5.1.2 Performance of Sample Meridian Filter 19
5.1.3 Performance of Sample Myriad Filter 20
5.1.4 Performance of WLS Filter 21
5.2 Cross-Comparison of Filters for Different Alpha Values 22
5.2.1 Cross-Comparison for $\alpha = 1.9$ 22
5.2.2 Cross-Comparison for $\alpha = 1.8$ 23
5.2.3 Cross-Comparison for $\alpha = 1.7$ 24
5.2.4 Cross-Comparison for $\alpha = 1.6$ 25
5.2.5 Cross-Comparison for $\alpha = 1.5$ 26

CHAPTER SIX - DETECTING FALSE DATA INJECTION ATTACKS.....27

6.1 Bad Data Detection 28
6.1.1 Bad Data Definition 28
6.1.2 Bad Data Detection Techniques 29
6.1.2.1 Chi-Square Distribution 29
6.1.2.2 Normalized Residuals 30
6.2 False Data Injection Attack Detection 30
6.2.1 False Data Injection Model 31
6.2.2 Two-Sided CUSUM Algorithm for Detecting False Data Injection Attack 33
6.3 A False Data Injection Attack Scenario and Its Detection 36
6.4 Performance Tests of CUSUM for Alpha-Stable Distributions 40

CHAPTER SEVEN - CONCLUSIONS 44

REFERENCES 45

APPENDICES.....45

LIST OF FIGURES

Page

Figure 1.1 Distributed topology for the future smart grid.....2

Figure 2.1 Real time network model at transmission level.....4

Figure 2.2 Two port π-model. 6

Figure 2.3 Illustration of WLS in impulsive noise environment.....8

Figure 3.1 Process of median filter (N=5) 10

Figure 3.2 Illustration of median filter in impulsive noise environment 10

Figure 3.3 Illustration of meridian filter in impulsive noise environment.....11

Figure 3.4 Illustration of sample myriad filter mechanism.....12

Figure 3.5 Illustration of myriad filter in impulsive noise environment.....13

Figure 4.1 Density functions of symmetric distributions for different α values. 15

Figure 4.2 View of tails for $= 0.5,\ \alpha = 1,\ \alpha = 1.5,\ \alpha = 2.0$.....15

Figure 4.3 Non-isotropic complex $S\alpha S$ random variables ($\alpha = 1.8$).....17

Figure 4.4 Isotropic complex $S\alpha S$ random variables ($\alpha = 1.8$) 17

Figure 5.1 Performance of sample median filter for different α values 19

Figure 5.2 Performance of sample meridian filter for different α values.....20

Figure 5.3 Performance of sample myriad filter for different α values.....20

Figure 5.4 Performance of WLS filter for different α values 21

Figure 5.5 FLOE versus SDR for $= 1.9$ 22

Figure 5.6 FLOE versus SDR for $= 1.8$ 23

Figure 5.7 FLOE versus SDR for $= 1.7$ 24

Figure 5.8 FLOE versus SDR for $= 1.6$ 25

Figure 5.9 FLOE versus SDR for $= 1.5$ 26

Figure 6.1 Bad data illustration 28

Figure 6.2 X^2 probability density function 30

Figure 6.3 (a) An example signal. (b) S_i^+ is upper CUSUM. (c) S_i^- is lower CUSUM. 35

Figure 6.4 Illustration of attacking point, missed attack samples, and detection point 36
Figure 6.5 IEEE-14 Bus System 37
Figure 6.6 Detection ratio (%) versus threshold 39
Figure 6.7 False alarm ratio (%) versus threshold 39
Figure 6.8 ARL versus threshold 40
Figure 6.9 Detection ratio (%) versus threshold 41
Figure 6.10 False alarm ratio (%) versus threshold 42
Figure 6.11 ARL versus threshold 43

LIST OF TABLES

Page

Table 6.1 Detection, misdetection, and false alarm terms 32
Table 6.2 Power flow data..... 38

CHAPTER ONE
INTRODUCTION

Smart power grids have recently become a crucial research subject which is expected to integrate advanced power, communications, signal processing, control, and computing technologies in order to improve robustness and efficiency of the power networks. Whereas current electricity systems are based on one-way flow of energy and information from the source to the end user, a smart grid system provides two-way flow of energy and information throughout the system.

Structure of a power grid involving large interconnected power systems is composed of grids consisting of multiple subnets and this complicated network is managed by an operator. The system operator needs more reliable and robust information about whole power grid to take appropriate precautions in failure or restricted conditions. Transmission system is under stress. Because generation and loading are increasing and capacity of transmission lines isn't increased sufficiently. Hence, transmission system must be operated at its maximum capacity in some cases. Hence, state estimation plays a crucial role to serve the state of the grid and enables energy management systems to perform various important control and planning tasks such as establishing near real-time network models for the grid, optimizing power flow, and bad data-injection detection (Huang, Werner, Huang, Kashyap, & Gupta, 2012).

Supervisory Control and Data Acquisition (SCADA) systems are used for real-time monitoring and controlling large-scaled power grid by system operator. SCADA provides a lot of information to operator like power flows, circuit-breaker positions, transformer taps, bus voltages, etc. Some faulty sensors and lost data could exist when transmitting data between Remote Terminal Units (RTUs). State estimator filters these errors for providing best estimated state to energy management system (EMS).

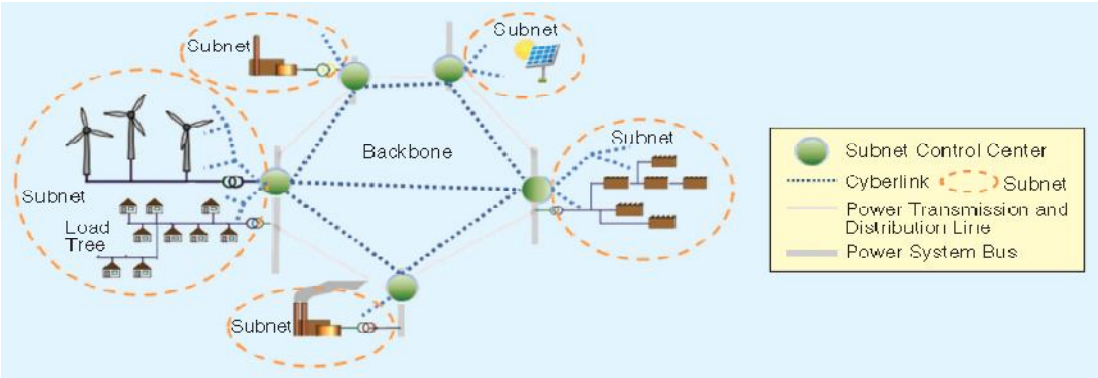

Figure 1.1 Distributed topology for the future smart grid (Cui et al., 2012).

Distributed future topology for future smart grid is illustrated in Figure 1.1. The future smart grids are expected to provide real-time system-wide state awareness. State estimation is a key function in building a real-time network model in the energy management system. State estimation refers to the procedure of obtaining the voltage magnitudes and phase angles at a bus which is located in the power grid. In literature, most of studies use weighted least squares (WLS) method for state estimation process and assume that the noise has Gaussian distribution. In this thesis, the robust M-filters are proposed for state estimation for non-Gaussian environments.

Gaussian distributions have been widely accepted as a tractable model in signal processing. But in real world applications, the processes are impulsive in nature, and are not well represented by Gaussian distributions (Arce, 2005). Hence, we discussed alpha-stable distributions and implemented our experiments for alpha-stable noise environments.

Measurements which are used for state estimation may contain errors that affect the accuracy of state estimation, named bad data, because of device failure, device misconfiguration, telecommunication medium, or other reasons. Identification and suppression of bad data is based on the state estimation method. Phasor measurement units (PMUs) can be used for defending against bad data in order to improve state estimation performance. Measurements collected by PMUs are synchronized by Global Positioning System (GPS). Thus, using PMUs in power grid improves the robustness of state estimation and bad data detection (Korres et al., 2011).

Conventional bad data detection techniques depend on looking gross errors which appear in measurement residuals. But these techniques are weak for catching highly structured bad data which is called false data injection attacks (Cui et al., 2012). Attacker can mislead the control center by injecting malicious data on state estimation process without being detected. In other words, attacker can obtain unauthorized information and use this information. Hence, operator could make wrong decision which causes electric power blackout in large area, economical issues, danger for electrical device equipment, etc. Because of these reasons, false data injection attacks to smart grid must be detected as quickly as possible for smart grid security. Speed of the detection of any malicious attack has a vital importance to enable defence strategies in a moderate time in the grid. The delay between attacking time and detection time should be as small as possible. This type of problem is called quickest detection problem.

Quickest detection algorithm tries to detect change as quickly as possible based on real time measurements when pre-defined conditions are met. Pre-defined conditions refer to the decision rules that optimize the trade-off between the detection speed and detection reliability (Huang, Werner, Huang, Kashyap, & Gupta, 2012). The CUSUM technique is presented in this thesis for quickest detection.

The thesis consists of seven chapters and the remainder of the thesis is organized as follows. Chapter Two outlines the state estimation in power grids and traditional state estimation method which is named weighted least squares (WLS). Chapter Three presents median, meridian and myriad filters which are called the robust M-filters and behavior of these filters in impulsive environments. Chapter Four includes general information about alpha-stable distributions. The performance tests of filters are simulated in MATLAB for comparing state estimation performances of WLS and robust m-filters under alpha-stable noise. The results are presented in Chapter Five. In Chapter Six, false data injection attack is defined, and CUSUM algorithm and performance of attack detection is presented. Performance tests are implemented in MATLAB environment, and results are shown in this chapter. Finally, conclusions are given in Chapter Seven.

CHAPTER TWO
STATE ESTIMATION IN POWER GRID

State estimation has an important role in supervisory control and planning of electric power grid (Huang, Werner, Huang, Kashyap, & Gupta, 2012). Because of the complexities of operating large and interconnected networks, Energy Management System (EMS) needs reliable information about power grid. EMS uses state estimation to process real time data which is collected by Supervisory Control and Data Acquisition (SCADA) system. In Figure 2.1, the mechanism of real time network model at transmission level is demonstrated. The goal of EMS is to monitor, control, analyze, plan, and optimize electric transmission lines and electric equipment like transformers, circuit breakers and generators.

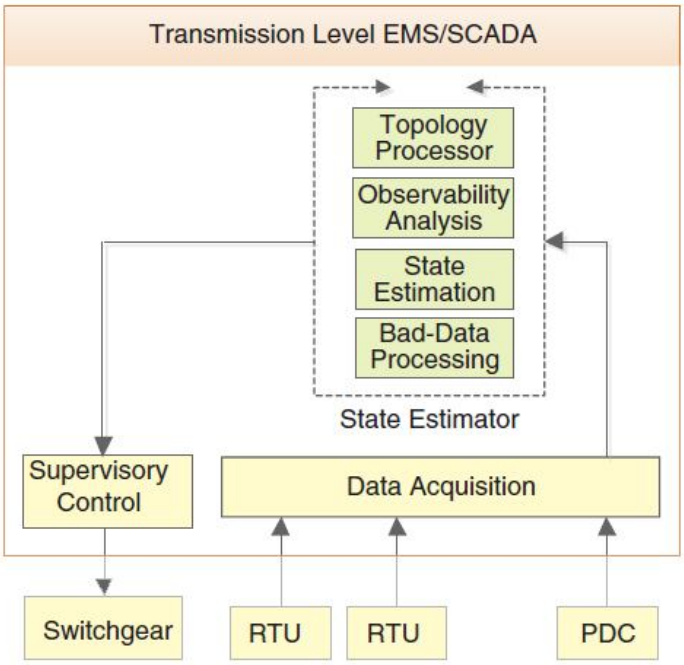

Figure 2.1 Real time network model at transmission level (Huang et al., 2012).

State estimation aims to get the best estimate of the current system state by processing sets of measurements and parameters for providing correct information about power grid to system operator. The success of the state estimation depends on the accuracy of measured data and network parameters. The measured data may be erroneous because of noise and error existing in communication or metering system.

Hence, state estimator works for filtering all these errors to achieve correct state of electric power grid in near real-time. WLS approach is widely used for state estimation. Static state estimation is only discussed in our approach. Dynamic state estimation is not included because of computational complexity.

2.1 Static State Estimation

Static state estimation refers to the procedure of obtaining the voltage magnitudes and phase angles at all buses which are located in the power grid at a given point in time (Abur & Exposito, 2004). Voltage magnitudes and phase angles are named state variables. It is assumed that the network topology and parameters are perfectly known, power system operates under balanced conditions, and measurement errors are independent. Another assumption is that noise has Gaussian distributions.

In an N-bus system, the $(2N-1) \times 1$ state vector has the form $x = [\theta_2, \theta_3, \dots, \theta_i, |V_1|, \dots, |V_i|]^T$ where θ_i represents phase angels and $|V_i|$ represents the magnitudes of the voltages at the ith bus. A bus is arbitrarily selected as a reference bus. Phase angle (θ_1) at reference bus is set to zero radians. A set of measurements $z \in \mathbb{R}^{L\times 1}$, $L > 2N-1$, is collected for estimating the state. Equation 2.1 shows the measurement vector:

$$z = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_i \end{bmatrix} = \begin{bmatrix} h_1(x_1, x_2, \dots, x_j) \\ h_2(x_1, x_2, \dots, x_j) \\ \vdots \\ h_i(x_1, x_2, \dots, x_j) \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \\ \vdots \\ n_i \end{bmatrix} = h(x) + n \qquad (2.1)$$

Specifically, $h_i(x)$ is the nonlinear function relating measurement i to state vector x, $x^T = [x_1, x_2, \dots, x_j]$ is the system state vector, $n^T = [n_1, n_2, \dots, n_j]$ is a zero-mean Gaussian measurement noise vector with covariance matrix $C_n \in \mathbb{R}^{L\times L}$.

The measurements may include active and reactive power flows, and bus power injections, line current flow magnitudes and voltage magnitudes at the buses. These measurements can be expressed in terms of voltage magnitudes and phase angles

based on two-port π-model for network branches as shown in Figure 2.2 (Abur & Exposito, 2004).

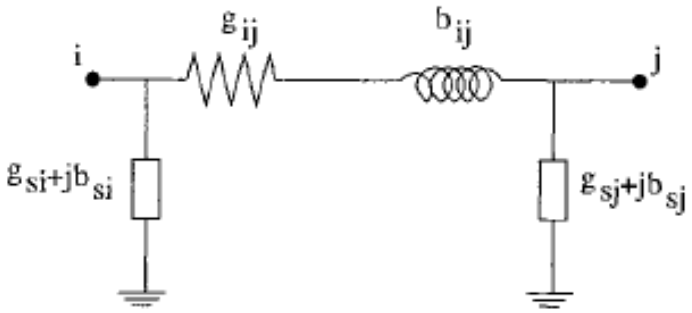

Figure 2.2 Two port π-model.

Specifically, $G_{ij} + jB_{ij}$ is ij th element of complex admittance matrix. θ_{ij} is $\theta_i - \theta_j$. V_i and V_j are voltage magnitude corresponding to i th and j th bus, respectively. $g_{ij} + jb_{ij}$ is admittance of the branch between bus i and bus j . $g_{si} + jb_{si}$ and $g_{sj} + jb_{sj}$ are admittance of the shunt branch.

- Real and reactive power injection at a bus i is expressed as follows:

$$\begin{aligned} P_i &= V_i \sum_{j \in \aleph_i} V_j \, (G_{ij} \cos\theta_{ij} + B_{ij} \sin\theta_{ij}) \\ Q_i &= V_i \sum_{j \in \aleph_i} V_j \, (G_{ij} \sin\theta_{ij} - B_{ij} \cos\theta_{ij}) \end{aligned} \tag{2.2}$$

- Real and reactive power flow from bus i to bus j is expressed as follows:

$$\begin{aligned} P_{ij} &= V_i^2 (g_{si} + g_{ij}) - V_i V_j \, (g_{ij} \cos\theta_{ij} + b_{ij} \sin\theta_{ij}) \\ Q_{ij} &= -V_i^2 (b_{si} + b_{ij}) - V_i V_j \, (g_{ij} \sin\theta_{ij} - b_{ij} \cos\theta_{ij}) \end{aligned} \tag{2.3}$$

- Line current flow magnitude from bus i to bus j is expressed as follows:

$$I_{ij} = \sqrt{(g_{ij}^2 + jb_{ij}^2)\,(V_i^2 + V_j^2 - 2\,V_i V_j \cos\theta_{ij})} \tag{2.4}$$

2.2 Weighted Least Squares (WLS)

WLS method is traditionally used to estimate the state vector in Gaussian environments from the measurement equation in 2.1. State estimation problem is solved by finding $\hat{x}$ as in the following equation:

$$\hat{x} = \arg\min_{x} [z - h(x)]^T W^{-1} [z - h(x)] \tag{2.5}$$

W is the weighting matrix which has diagonal elements related to noise covariance. W is diagonal because of the independence of measurement errors. $\hat{x}$ is solved in an iterative way by linearizing Equation 2.1 around the available estimate and Gauss-Newton algorithm is applied for improving performance of the estimation (Huang, Werner, Huang, Kashyap, & Gupta, 2012).

2.3 Non-Gaussian and Impulsive Noise Approach

Traditional state estimation assumes that noise has Gaussian distribution as mentioned before. However, in practical applications, measurement and process noise have non-Gaussian distribution. If noise distribution is non-Gaussian, performance of WLS method will be dramatically decreased. M-filters provide better performance than linear filters in non-Gaussian noise environments (Pander & Przybyła, 2012).

Noise processes in practice are generally impulsive in nature and are not well described with Gaussian distribution (Arce, 2005). The impulsive noise may cause false operations if it is not handled appropriately. Hence, impulsive noise is needed to be suppressed for accurate analysis. The impulsive noise in our approach is modeled by alpha-stable distributions which is well-suited for describing impulsive events. The robust M-filters can be applied in different types of digital signal processing applications like impulsive environments (Pander & Przybyła, 2012). As seen in Figure 2.3, WLS cannot suppress impulsive components. M-filters like median, meridian and myriad give better results than WLS approach under impulsive noise. These filters are discussed in detail in the next chapter.

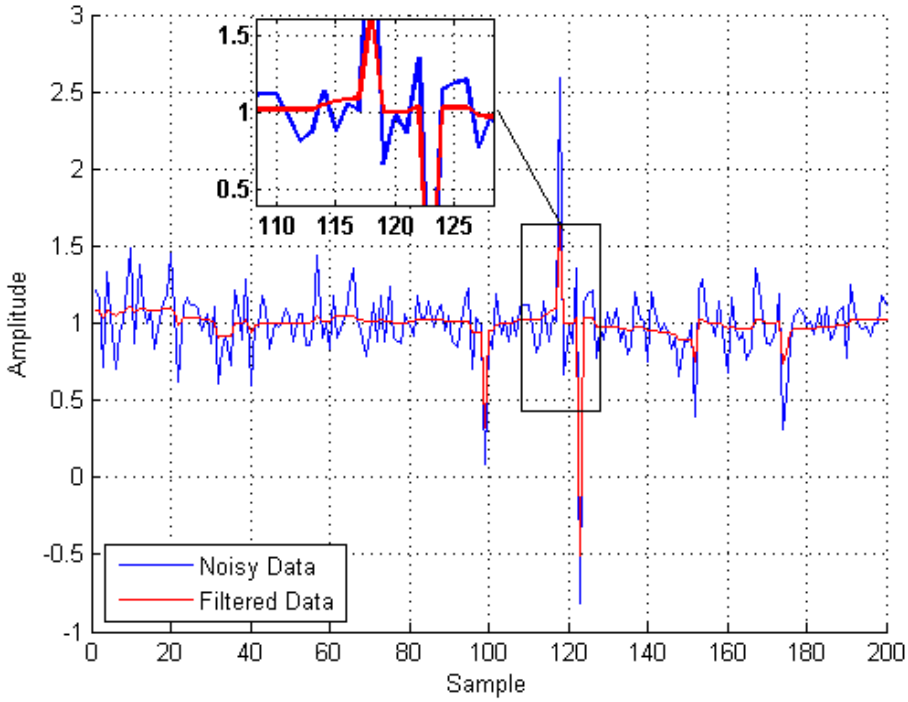

Figure 2.3 Illustration of WLS in impulsive noise environment.

CHAPTER THREE
ROBUST M-FILTERS FOR STATE ESTIMATON

In recent years, filtering process based on M-estimators which are also named robust filters is widely used in signal processing. The median, meridian and myriad filters are type of robust M-estimators which are very useful for suppressing impulsive noise (Pander & Przybyła, 2012).

3.1 Robust M-Filters

Formulation of M-estimators is shown in following way. A set of i data samples $x_{1,}x_{2,}\dots,x_i$ is given, where $x_i = \beta_i + v_i$ and $1 \leq i \leq N$. β_i is location parameter which is needed to be estimated under noise v_i. Distribution of noise is not exactly known. The only assumption is that noise has symmetric, independent, identical distribution (symmetric i.i.d.) (Pander & Przybyła, 2012).

The M-estimate of $\hat{\beta}$ is shown as a minimum global energy function in the following expression:

$$\hat{\beta} = \arg\min_{\beta \in \Re} \sum_{i=1}^{N} \rho(x_i - \beta) \tag{3.1}$$

Specifically, $\rho(.)$ is called the cost function, and $\hat{\beta}$ is location parameter of M-estimator which minimizes the expression in Equation 3.1. M-estimator's behavior is totally characterized by the shape of the cost function (Pander & Przybyła, 2012).

3.2 Sample Median Filter

To define the median filter, let $X[.]$ be the discrete time signal. Median filter passes a window over the signal $X[.]$ that selects, at each instant n, an odd number of sequential samples to comprise the observation vector $X[n]$. $X[n] = [\,X[n - N_L], \dots, X[n], \dots, X[n + N_R]]^T$ is the observation window which is centered at n, and $N = N_L + N_R + 1$ is the window length which is selected based on input data. Generally, observation window is symmetric ($N_L = N_R$) (Arce, 2005).

The median filter sorts samples which is located in the observation window, selects the median or middle value from the sorted window and produces the output signal, defined at time index n. Median filter can be expressed as below.

$$Y[n] = MEDIAN\,[\,X_1[n], \ldots, X_N[n]\,] \tag{3.2}$$

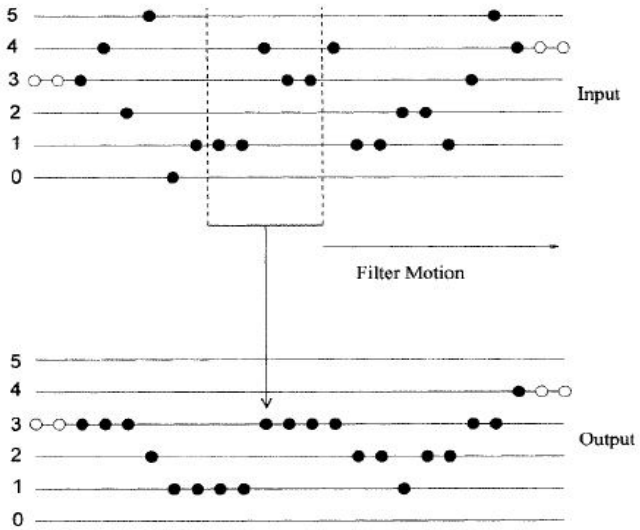

Figure 3.1 Process of median filtering (N=5) (Arce, 2005).

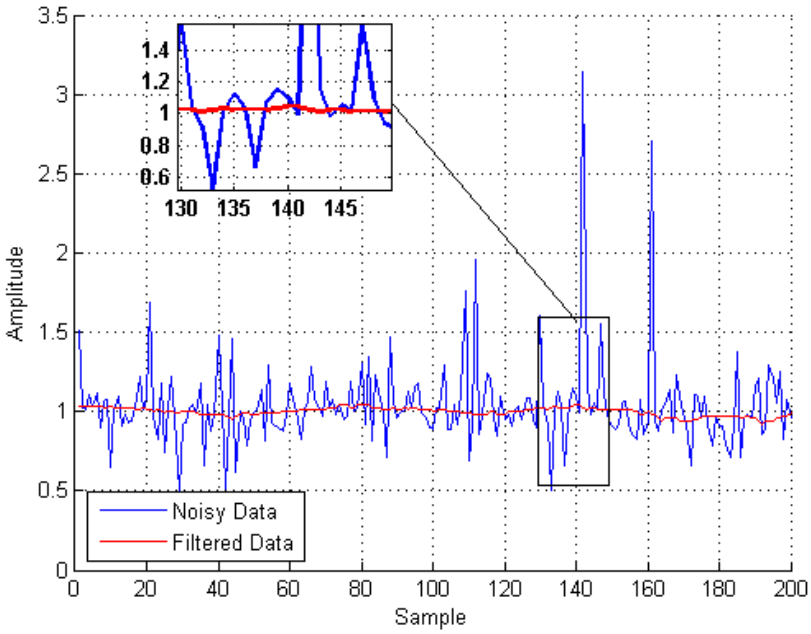

Figure 3.2 Illustration of median filter in impulsive noise environment.

Figure 3.1 shows how median filter works on data. In Figure 3.2, it can be seen that median filter can suppress impulsive noise components and give more reliable results.

3.3 Sample Meridian Filter

Meridian distribution is defined as a random variable formed as the ratio of two independent zero-mean Laplacian distributed random variables. A set of N independent and identically distributed samples $x_1, x_2, \ldots, x_N$ each obeying the meridian distribution with common scale parameter δ which is called medianity parameter, the sample meridian $\hat{\beta}$ is given by the following equation (Aysal & Barner, 2007):

$$\hat{\beta} = \arg\min_{\beta \in \Re} \sum_{i=1}^{N} \log\left[\delta + |x_i - \beta|\right] = meridian\left\{x_i|_{i=1}^{N} ; \delta\right\} \tag{3.3}$$

where β is the location parameter. Sample meridian includes the free-tunable parameter δ which plays an important role in the behavior of meridian estimator, unlike sample mean and median. If medianity parameter tends to lower values, the estimator becomes more robust against the impulsive noise (Aysal & Barner, 2007).

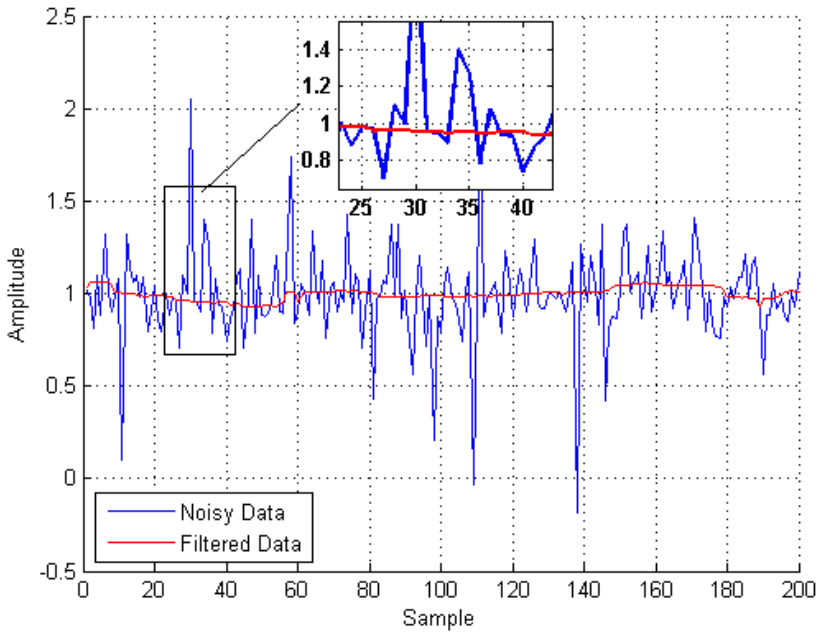

Figure 3.3 Illustration of meridian filter in impulsive noise environment.

In Figure 3.3, performance of the meridian filter is good enough for suppressing the impulsive noise. The robustness of meridian filter can be improved by tuning the parameter $\boldsymbol{\delta}$.

3.4 Sample Myriad Filter

Myriad filtering is based on maximum likelihood estimate of location under Cauchy distribution. For a set of N i.i.d. samples $x_1, x_2, \ldots, x_N$ each obeying the Cauchy distribution with common scale parameter K which is called linearity parameter, the sample myriad $\hat{\beta}$ can be computed using the following equation (Pander & Przybyła, 2012):

$$\hat{\beta} = \arg \min_{\beta \in \Re} \sum_{i=1}^{N} \log [K^2 + |x_i - \beta|^2] = myriad \{x_i|_{i=1}^{N} ; K\} \tag{3.4}$$

where β is location parameter. The influence of gross errors or outliers is de-emphasized by the logarithm function.

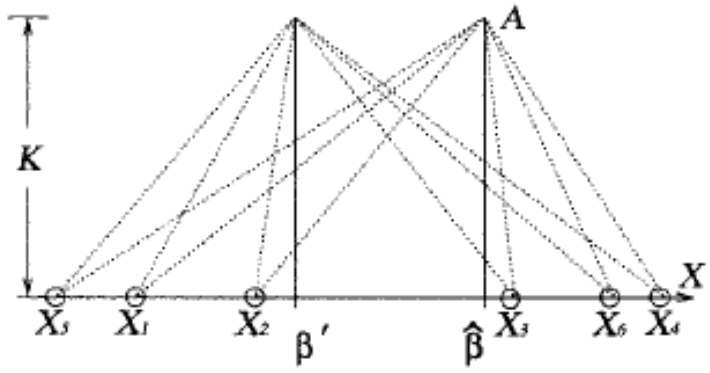

Figure 3.4 Illustration of sample myriad filter mechanism.

Myriad filter working principle is shown in Figure 3.4. $\hat{\beta}$ minimizes the product of distances from point A to all samples. Linearity parameter K controls robustness of the myriad filter (Pander & Przybyła, 2012). When K is large, the robustness of myriad filter is low, and most of the sample data is taken into consideration for estimating location. Conversely, if K decreases, resistance to outliers increases. The parameter of K is determined by the impulsiveness of the noise process. Hence, the

degree of impulsiveness of noise is firstly determined through estimating the stability parameter, α. This subject is out of scope for this thesis. K can be calculated by the following formula (Gonzales, Griffith, & Arce, 1996).

$$K(\alpha,\gamma) = \gamma^{1/\alpha}\tan\left(\pi\frac{\alpha}{4}\right) \tag{3.5}$$

Specifically, α is characteristic exponent parameter, and γ is dispersion parameter of distribution. These parameters will be discussed in detail in the next chapter.

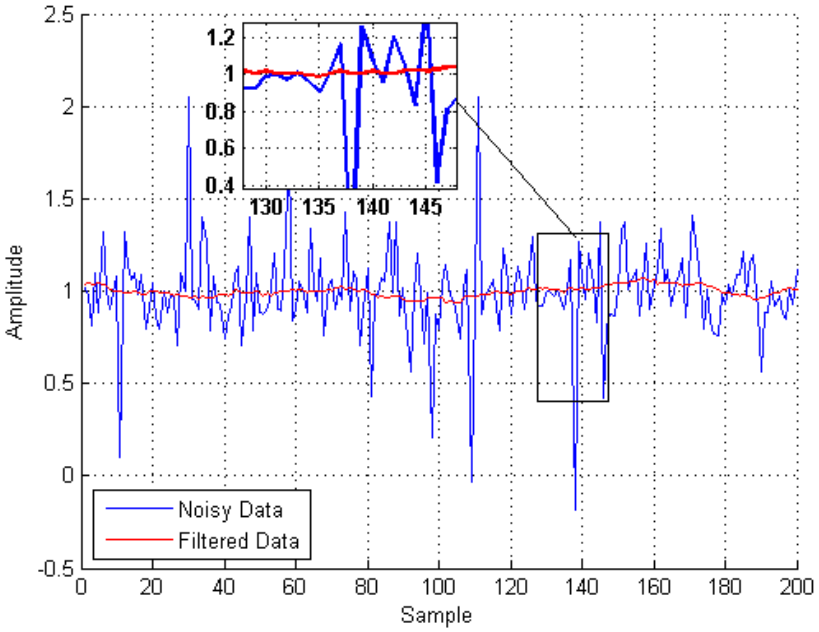

Figure 3.5 Illustration of myriad filter in impulsive noise environment.

Figure 3.5 shows that sample myriad filter can successfully suppress the impulsive noise. The K parameter can be tuned for more robustness.

CHAPTER FOUR
ALPHA STABLE DISTRIBUTIONS

Gaussian distributions have been widely accepted as a useful and mathematically tractable model in signal processing. But in real world applications, the processes are impulsive in nature, and are not well represented by Gaussian distributions (Arce, 2005). Hence, in this section, we introduced a statistical model relying on symmetric alpha-stable ($S\alpha S$) distributions which are capable of describing impulsive signals in nature.

4.1 Alpha-Stable Distributions in Real Form

The characteristic function of alpha-stable distributions is written as follows:

$$\varphi(t;\delta,\gamma,\alpha,\beta)=\begin{cases}\exp\left[\, it\delta-|\gamma t|^{\alpha}\,\left(1-i\frac{2}{\pi}\beta sgn(t)\log|t|\right)\right] & for\quad \alpha=1\\ \exp\left[\, it\delta-|\gamma t|^{\alpha}\,\left(1-i\beta sgn(t)\tan\left(\pi\frac{\alpha}{2}\right)\right)\right] & for\quad \alpha\neq 1\end{cases} \tag{4.1}$$

The alpha-stable distribution is controlled by four parameters and is usually denoted by $S(\alpha,\beta,\gamma,\delta)$. Alpha, $\alpha \in (0,2]$, is a crucial parameter which is named characteristic exponent. Alpha describes the shape of the distribution. Beta, $\beta \in [-1,1]$, is skewness parameter which determines if the distribution is right or left skewed. Gamma, $\gamma > 0$, is named dispersion which determines the spread of distribution which is similar to variance of Gaussian distributions. Delta, $-\infty < \delta < +\infty$, is location parameter.

There are three special cases for α-stable distributions. For $\alpha = 2$, the distribution is named Gaussian distribution with variance $\sigma^2 = 2\gamma^2$, and the skewness parameter has no importance. For $\alpha = 1$ and $\beta = 0$, the distribution is named Cauchy distribution with scale parameter γ and location parameter δ. For $\alpha = 0.5$ and $\beta = 1$, it is named Lévy distribution with scale parameter γ and location parameter δ.

For symmetric approach, $\beta = 0$, the characteristic function turns into following expression:

$$\varphi(t;\delta,\gamma,\alpha)=\exp[\, it\delta-|\gamma t|^{\alpha}\,] \tag{4.2}$$

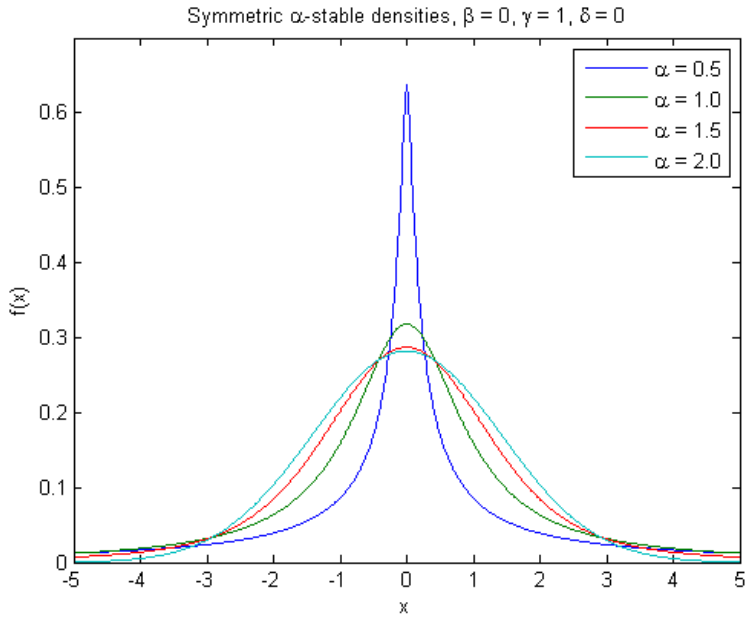

Figure 4.1 Density functions of symmetric distributions for different α values.

The symmetric density functions for different α values are plotted in Figure 4.1. The smaller alpha values cause the heavier tails of the $S\alpha S$. This means that alpha-stable distributions with small alpha values better represent highly impulsive signals or noises. On the other hand, if α decreases, the existence rate and the strength of the outliers increase (Samoradnitsky & Taqqu, 1994). Tails of distributions can be clearly seen in Figure 4.2.

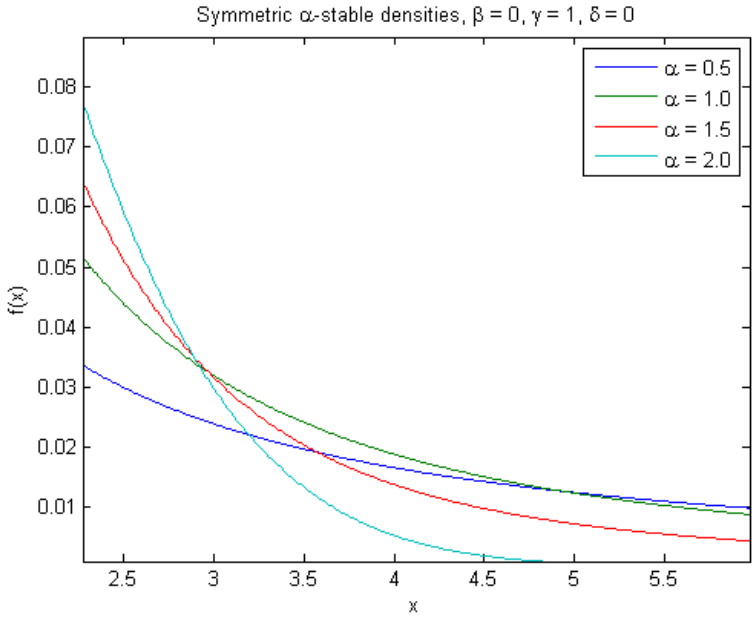

Figure 4.2 View of tails for $\alpha = 0.5,\ \alpha = 1,\ \alpha = 1.5,\ \alpha = 2.0$.

4.2 Alpha-Stable Distributions in Complex Form

During the state estimation process, we need to find complex bus voltage at each bus. Hence, it is needed to add complex noise on raw complex bus voltage for a more realistic approach. Because of this reason, we introduced complex $S\alpha S$ random variables.

Let X_1 and X_2 be real random variables which are defined on the same probability space. The complex random variable $X = X_1 + iX_2$ is shaped by the joint distribution of X_1 and X_2. If random vector (X_1, X_2) is symmetric alpha-stable in $\mathbb{R}^2$ then the complex random variable $X = X_1 + iX_2$ is named symmetric alpha-stable ($S\alpha S$) (Samoradnitsky & Taqqu, 1994).

If $e^{i\emptyset}X \overset{d}{=} X$ for any $\emptyset \in [0,2\pi)$, a complex random variable $X = X_1 + iX_2$ is said to be isotropic. For $\alpha = 2$, if X_1 and X_2 are i.i.d. then isotropy condition is satisfied. For $\alpha < 2$ condition, a complex random variable is isotropic if and only if (X_1, X_2) has a uniform spectral measure. Real and imaginary parts of an isotropic $S\alpha S$ random variable are dependent (Samoradnitsky & Taqqu, 1994).

There are some steps for generating complex isotropic $S\alpha S$ random variable as $X = X_1 + iX_2$. For $\alpha < 2$, we need to generate two i.i.d. zero mean normal random variables G_1 and G_2 and a random variable $A \sim S(\alpha/2,1,(cos\pi\alpha/4)^{2/\alpha}, 0)$ which is independent of (G_1, G_2). The vector (X_1, X_2) is sub-Gaussian with underlying vector (G_1, G_2). This implies that $(X_1, X_2) \overset{d}{=} (A^{1/2}G_1, A^{1/2}G_2)$. Every complex isotropic $S\alpha S$ random variable can be written as follows (Samoradnitsky & Taqqu, 1994).

$$X = A^{1/2}\,(G_1 + \, iG_2) \tag{4.3}$$

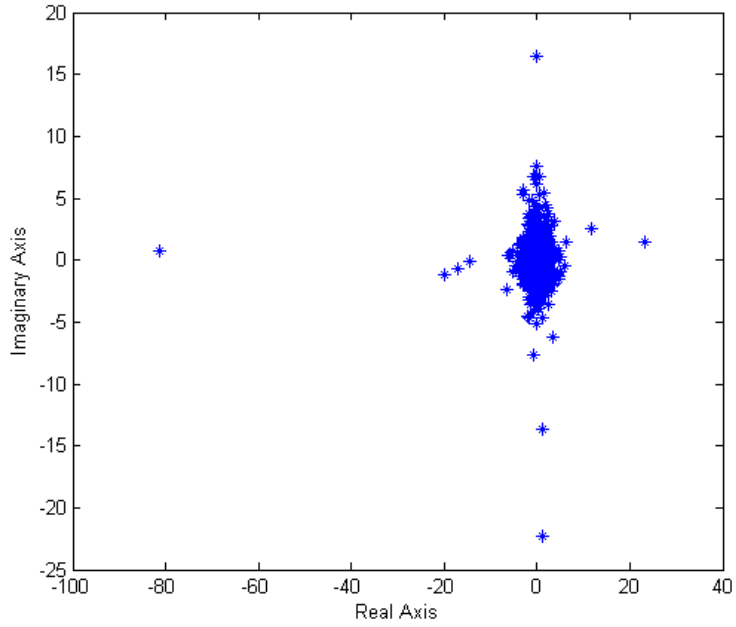

Figure 4.3 Non-isotropic complex $S\alpha S$ random variables ($\alpha = 1.8$).

In Figure 4.3, the data can be seen non-centralized for non-isotropic complex $S\alpha S$ random variables for $\alpha = 1.8$. If we generate isotropic complex $S\alpha S$ random variables, the data will be more centralized and have a circular behavior as shown in Figure 4.4.

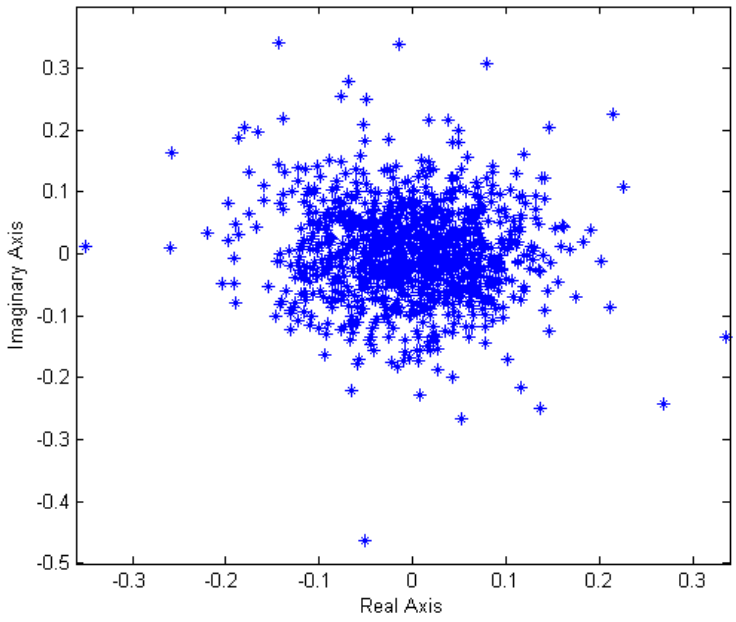

Figure 4.4 Isotropic complex $S\alpha S$ random variables ($\alpha = 1.8$).

CHAPTER FIVE
SIMULATIONS OF STATE ESTIMATION

State estimation is defined as a process of obtaining the voltage magnitudes and phase angles at all buses which are located in smart grid. Hence, we have generated raw bus voltage data through using MATPOWER (version 1.4) which is an open-source simulation package of MATLAB for solving power flow and optimal power flow problems (Zimmerman, Murillo-Sánchez, & Gan, 2011). MATPOWER also gives complex bus voltage values at each bus that we need, and includes IEEE 14, IEEE 30, IEEE 57, IEEE 118, and IEEE 300 bus system topologies and parameters.

In this section, simulations and results for impulsive and non-Gaussian ($1.5 \leq \alpha \leq 1.9$) environment have been presented for state estimation. Isotropic complex $S\alpha S$ noise is added on complex bus voltage data which is generated by MATPOWER. Data length is $N = 1000$ sample for each test. Alpha-stable noise sample is generated by an open-source toolbox which is referred in (Veillette, 2012). The number of individual runs is 200 for all filters at each dB point. Performances of filters are compared with each other and individually for $1.5 \leq \alpha \leq 1.9$ case. Observation window length is selected as 21 and symmetric for all robust filters mentioned before. Fractional lower order error (FLOE) versus signal to dispersion ratio (SDR) is used for performance tests. FLOE is expressed as follows.

$$FLOE = \frac{1}{N} \sum_{i=1}^{N} | x_i - \bar{x} |^p \text{ for } p < \alpha. \tag{5.1}$$

Specifically, x_i represents estimated state, $\bar{x}$ represents actual state and N represents data length. Signal to dispersion ratio (SDR) is defined as follows (Gonzales, Griffith, & Arce, 1996).

$$SDR = 20 \log_{10} \frac{A}{\gamma^{1/\alpha}\sqrt{2}}. \tag{5.2}$$

In the next subsection, results of the mentioned filters are presented with self-comparison and cross-comparison to interpret their performances in impulsive noise environments.

5.1 Self-Comparison of Filters for Different Alpha Values

5.1.1 Performance of Sample Median Filter

Performance of median filter with respect to characteristic exponent is shown in Figure 5.1. When noise becomes more impulsive, performance of the sample median filter decreases.

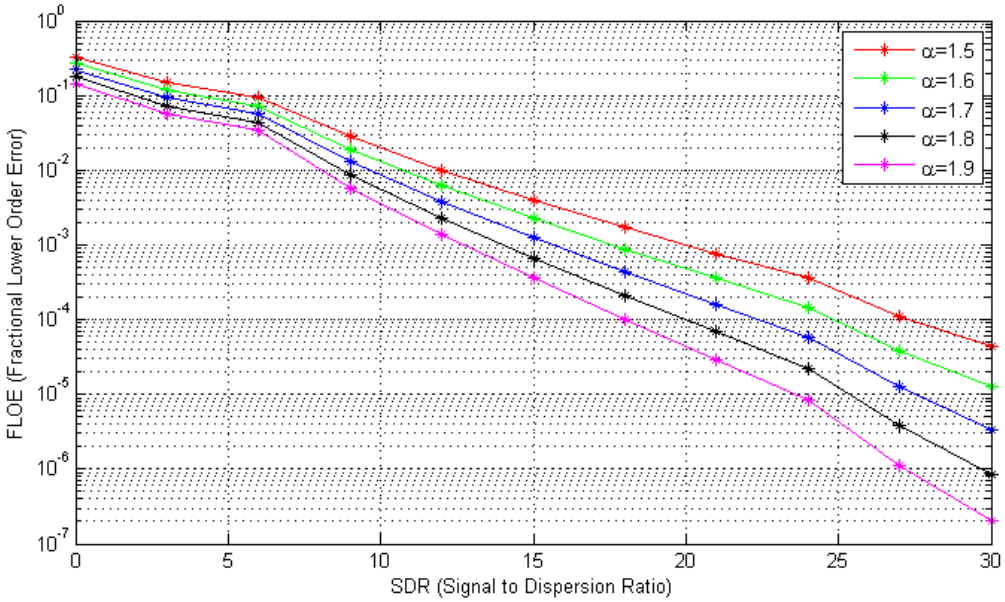

Figure 5.1 Performance of sample median filter for different α values.

Moreover, performance difference of median filter for different characteristic exponents grows with respect to increasing signal to dispersion ratio.

5.1.2 Performance of Sample Meridian Filter

The noise filtering effect for the same system using sample meridian filter is illustrated in Figure 5.2. Identically, the fractional lower order error slightly decreases proportionally with characteristic exponent. Differing from median filter, it can be observed that the performance saturates while signal to dispersion ratio increases. On the other hand, there is no such a growing performance difference with respect to characteristic exponent.

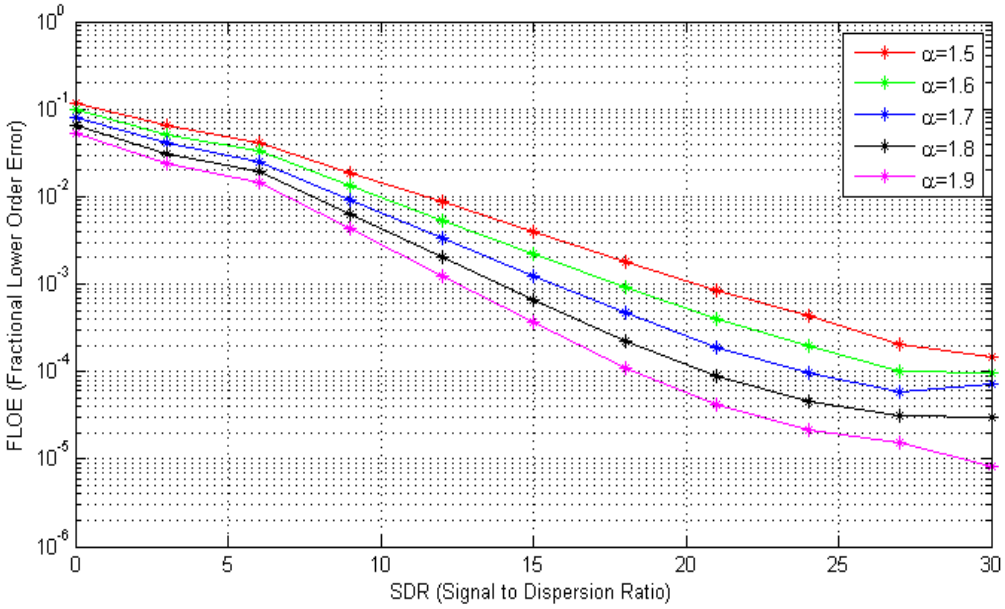

Figure 5.2 Performance of sample meridian filter for different α values.

5.1.3 Performance of Sample Myriad Filter

The simulations using sample myriad filter shown in Figure 5.3 represents that the impulsive behavior of the contaminated noise directly affects the de-noising capacity of the filter. Performance is quite close to the sample meridian filter.

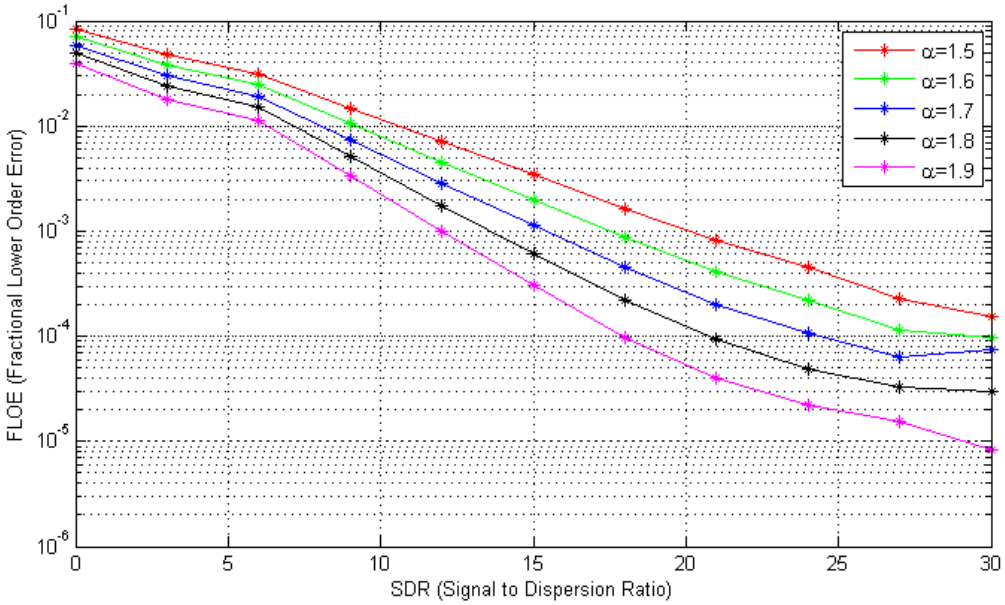

Figure 5.3 Performance of sample myriad filter for different α values.

5.1.4 Performance of WLS Filter

Although weighted least squares filter is suited for estimation under Gaussian disturbance, its performance was also measured under non-Gaussian noise model in order to compare with other filters. Results are shown in Figure 5.4. As signal to dispersion ratio increases performance of WLS filter also gets better. Interestingly, the noise filtering performance is strongly increased while the noise gets closer to Gaussian distribution rather than impulsive distribution.

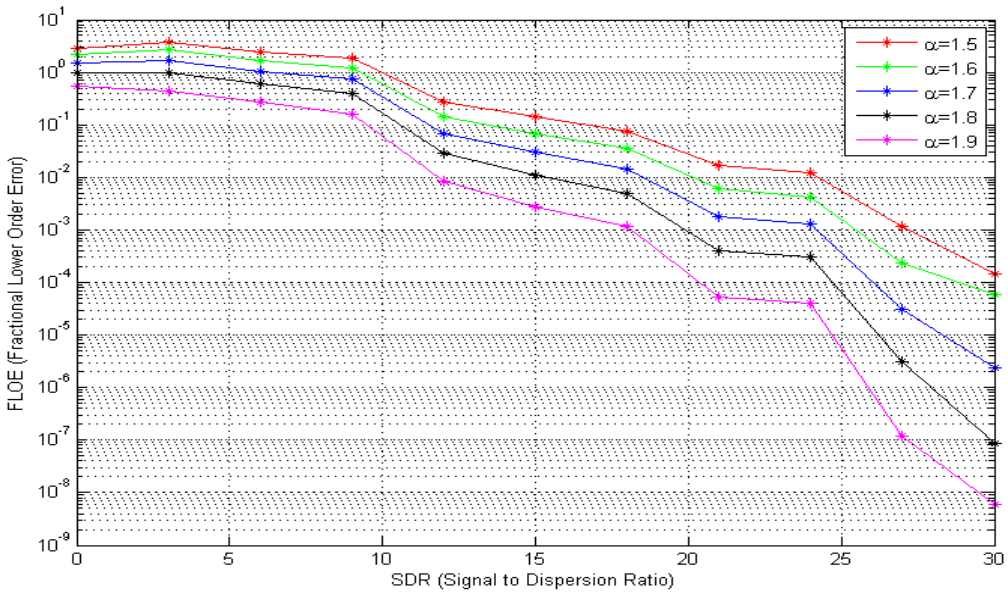

Figure 5.4 Performance of WLS filter for different α values.

One can say that all of the filter performances degrade when the characteristic exponent decreases, i.e., data becomes more impulsive. Namely, the filtering performance dramatically gets poorer when the corrupted noise becomes more impulsive.

In the next subsection, performances of mentioned filters will be compared while the characteristic exponent remains fixed.

5.2 Cross-Comparison of Filters for Different Alpha Values

In the previous section, the same filter was applied for noise contamination with different characteristic exponents. In this section, the cross comparison among the filters is represented by the FLOE versus signal to dispersion ratio.

5.2.1 Cross-Comparison for $\alpha = 1.9$

When the characteristic exponent has the value $\alpha = 1.9$ as shown in Figure 5.5, the noise is relatively close to Gaussian behavior. Sample myriad and sample meridian filters can be said to have almost the same performances. Although the sample median and weighted least square filters have poorer performances, starting from a certain SDR value, these two methods, especially WLS has a significant superiority in FLOE compared with the other filters.

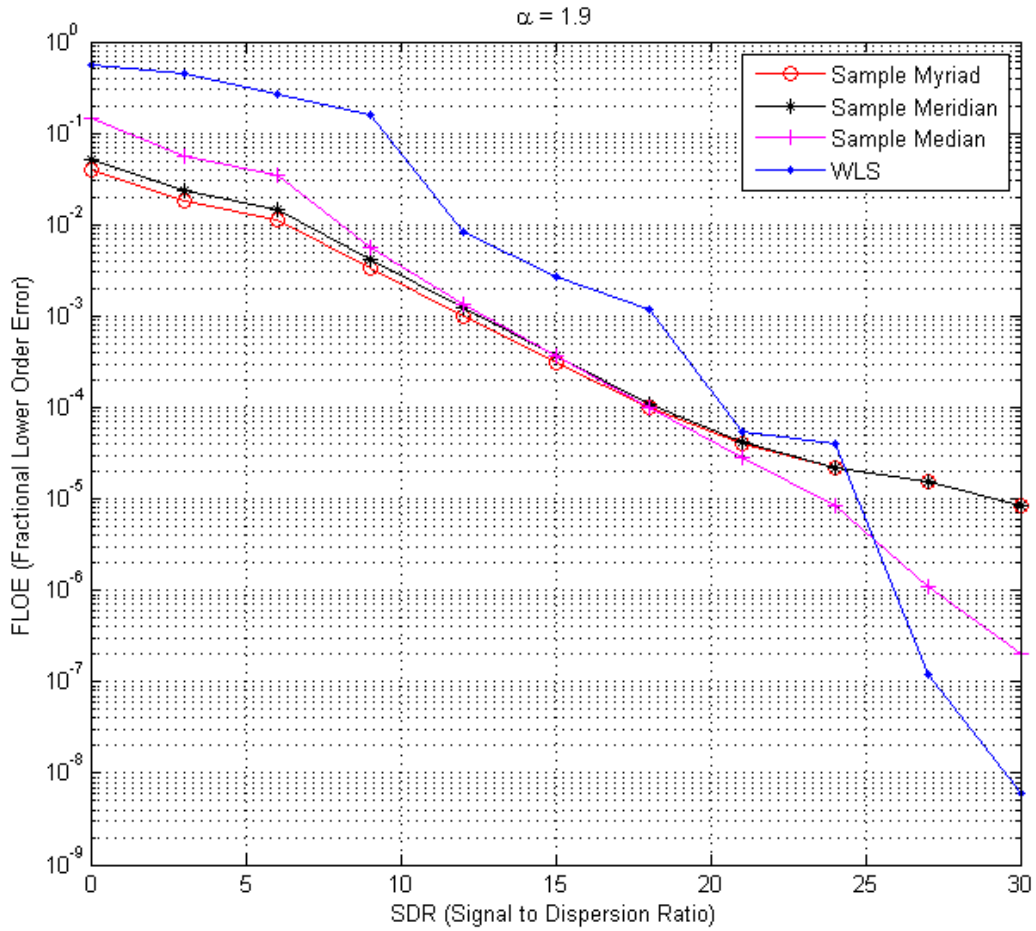

Figure 5.5 FLOE versus SDR for $\alpha = 1.9$.

5.2.2 Cross-Comparison for $\alpha = 1.8$

In this section the stable noise characteristic exponent is applied as $\alpha = 1.8$ as shown in Figure 5.6. The identical filter performances can be observed again. The only the difference is performances of all the filters become poorer as mentioned in the previous section.

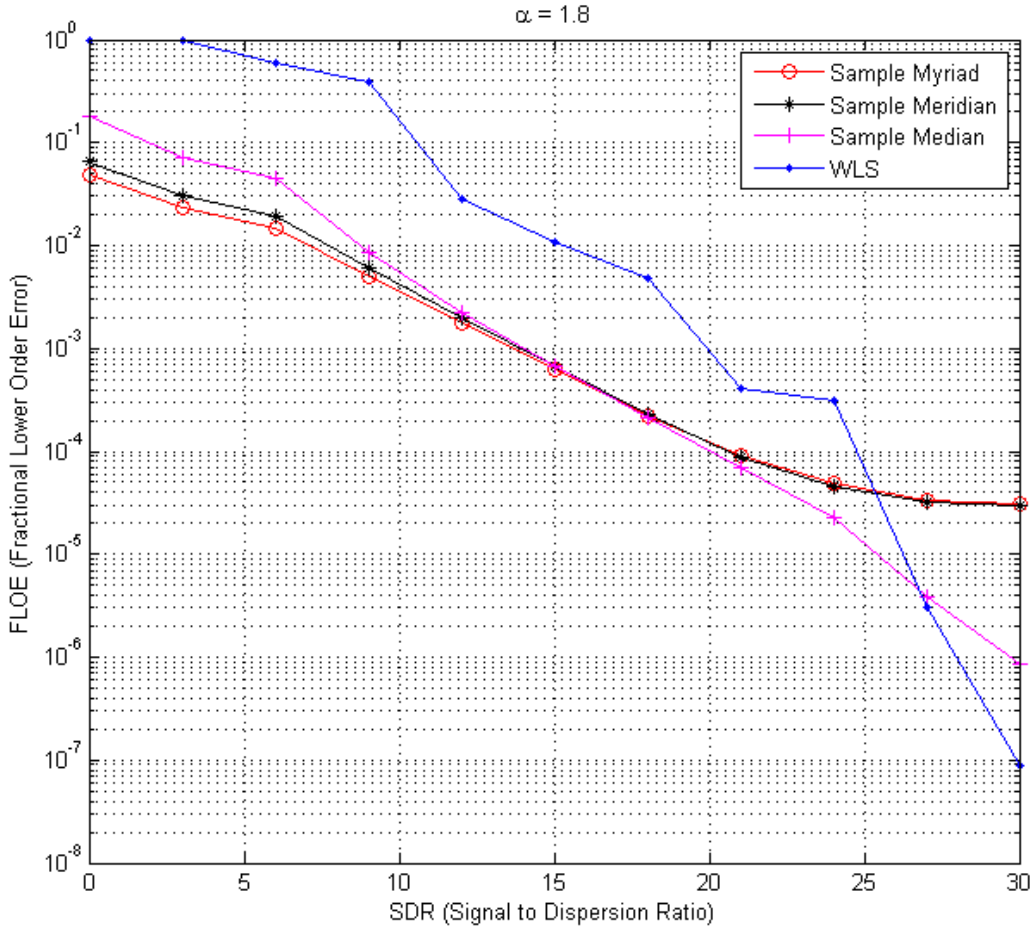

Figure 5.6 FLOE versus SDR for $\alpha = 1.8$.

5.2.3 Cross-Comparison for $\alpha = 1.7$

Decreasing the characteristic exponent yields a dramatic performance degradation in WLS filter, as illustrated in Figure 5.7. Other filters have superior performance than the WLS filter. Note that the sample median filter has the best error performance for the high SDRs while the sample myriad filter has better performance for low SDRs.

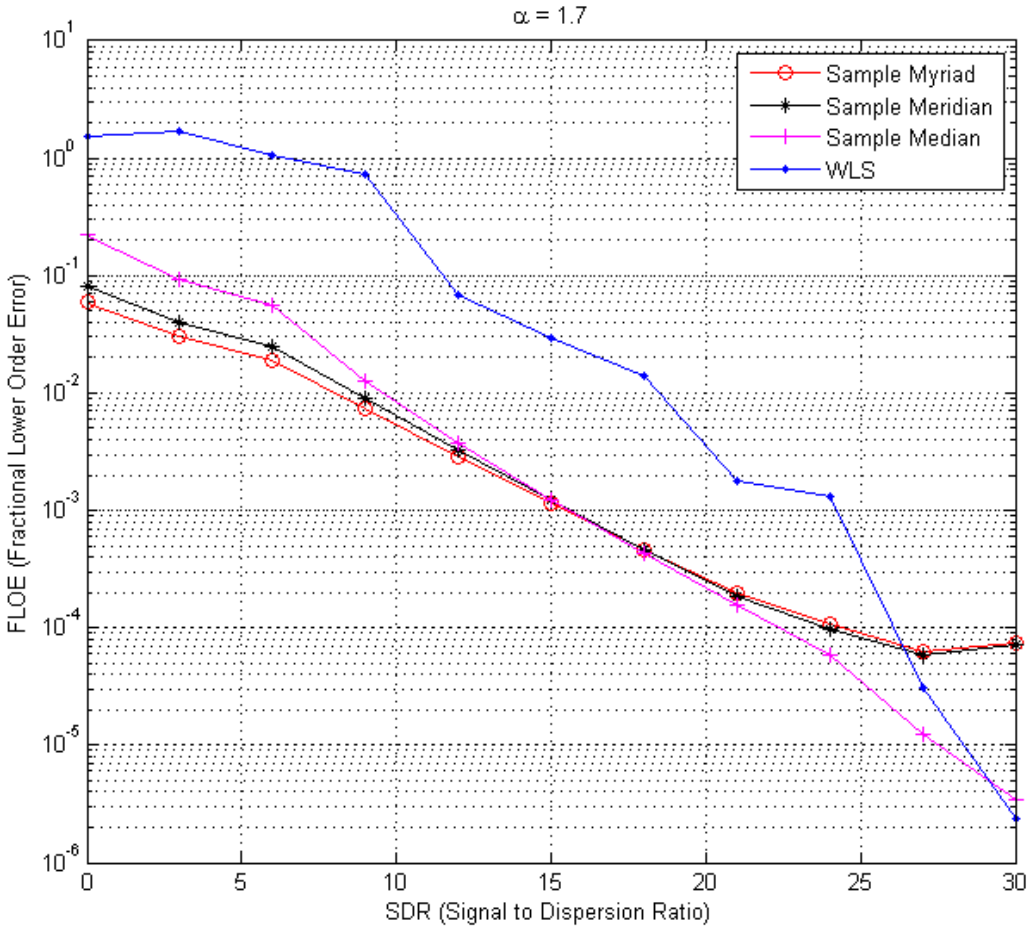

Figure 5.7 FLOE versus SDR for $\alpha = 1.7$.

5.2.4 Cross-Comparison for $\alpha = 1.6$

Differing from the previous characteristic exponent values, the WLS filter performances given in Figure 5.8 cannot exhibit a better performance in entire SDR interval. At the same time, one can observe that FLOE of sample median filter continuously decreases while sample myriad and sample meridian filters sature for large SDRs.

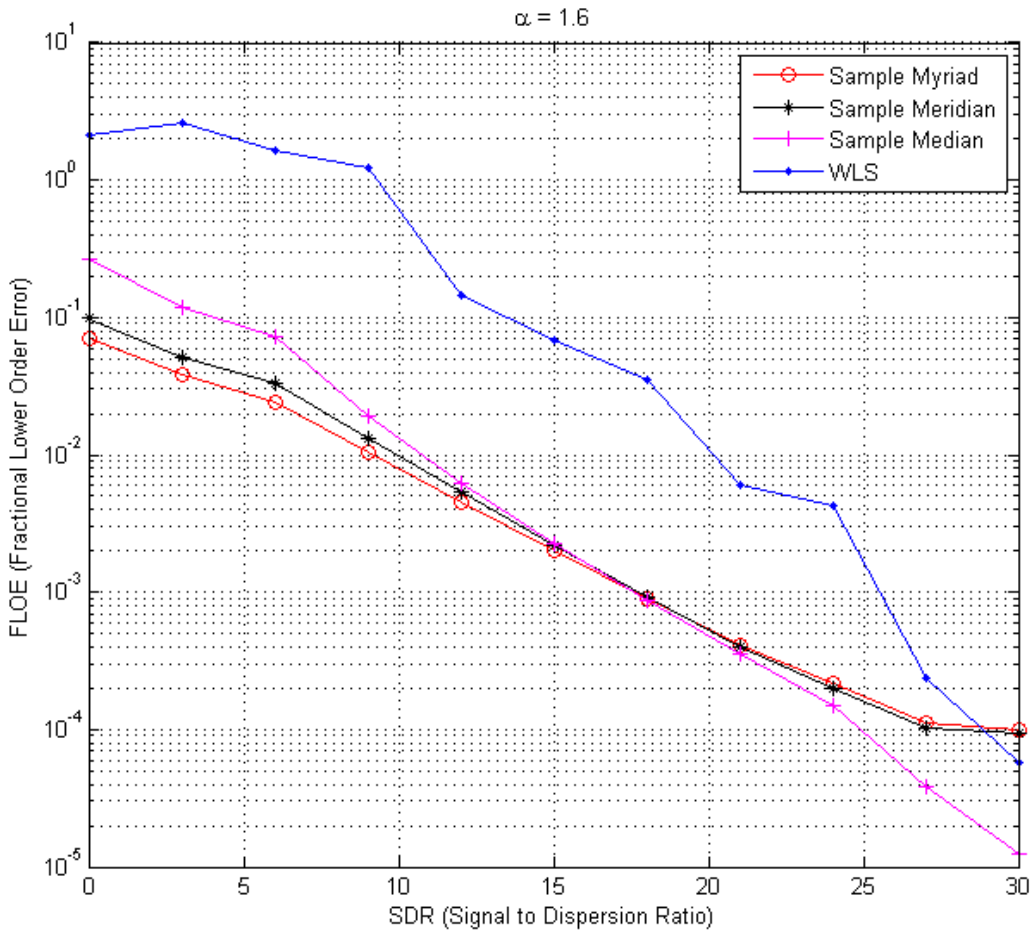

Figure 5.8 FLOE versus SDR for $\alpha = 1.6$.

5.2.5 Cross-Comparison for $\alpha = 1.5$

The heaviest impulsive noise involved in this thesis is $\alpha = 1.5$ for which sample myriad and sample meridian filters are superior for most of the SDRs. Only for a restricted interval of SDR, the sample median filter can have a better performance than the rest of the filters as shown in Figure 5.9.

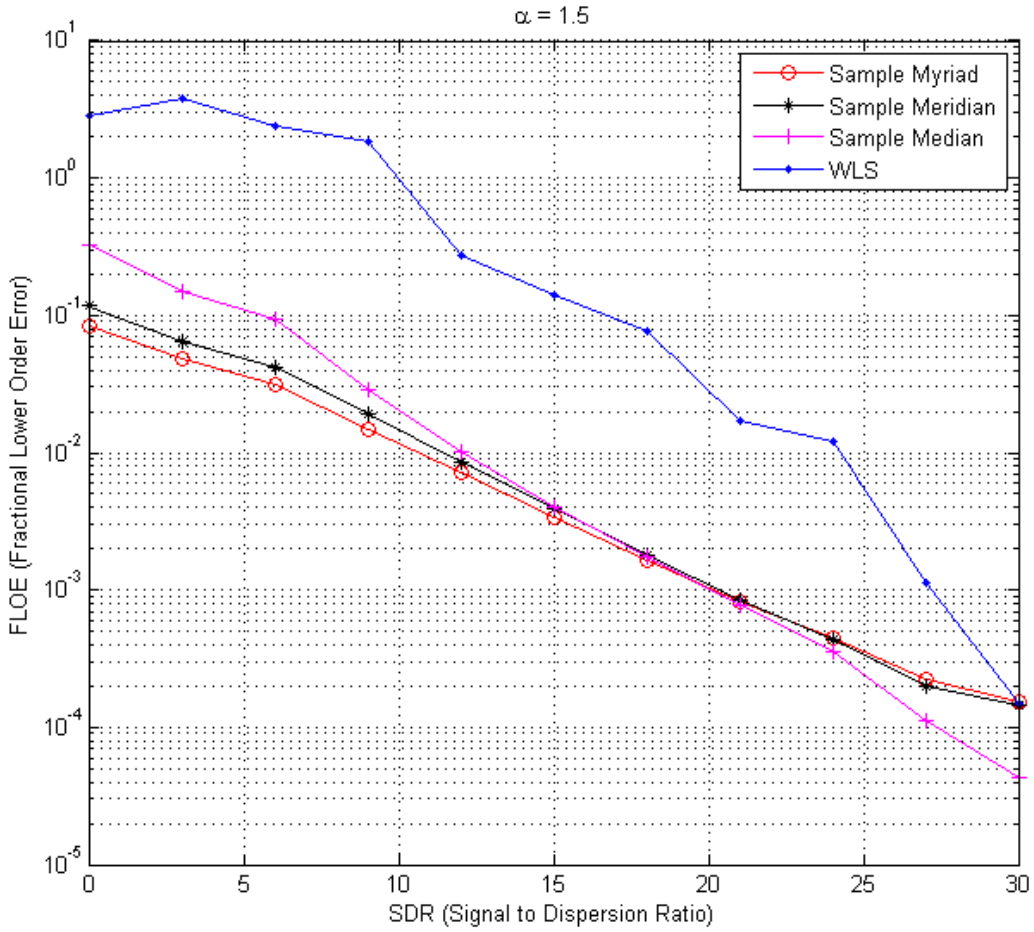

Figure 5.9 FLOE versus SDR for $\alpha = 1.5$.

CHAPTER SIX
DETECTING FALSE DATA INJECTION ATTACKS

The researchers increasingly emphasize the importance of smart grid, because smart grid supports clean, economic, and sustainable energy utilization. The robustness and efficiency of power grid is enhanced through using modern communication, signal processing, and control technologies with two-way communication. While smart grid makes power grid more intelligent, the risk of cyber attacks also increases.

The power grid is an interconnected system and spread out over a large geographical area. Supervisory control and data acquisition (SCADA) systems are used for monitoring and controlling large-scaled power grid by a system operator. SCADA provides a lot of information to operator like power flows, circuit-breaker positions, transformer taps, bus voltages, etc. Some faulty sensors and lost data could exist when transmitting data between RTUs. State estimator filters these errors for providing best estimated state to energy management system (EMS).

As mentioned earlier, state estimation process has a crucial role in supervisory control and planning of power grid. Measurements which are used for state estimation may contain errors that affect the accuracy of state estimation, named bad data, because of device failure, device misconfiguration, telecommunication medium, or other reasons. Identification and suppression of bad data is based on the state estimation method. Conventional bad data detection techniques depend on looking at gross errors which appear in measurement residuals. But these techniques are weak for catching highly structured bad data which is called false data injection attack. Attacker can mislead the control center by injecting malicious data on state estimation process without being detected. In other words, attacker can obtain unauthorized information and use this information to mislead EMS. Therefore, operator could make wrong decision which causes electric power blackout in a large area, economical losses, danger for electrical device equipment, etc. Because of these reasons, false data injection attacks to smart grid must be detected as quickly as possible.

For the attack detection problem, speed of the detection of any malicious attack has a vital importance to enable defence strategies in a moderate time in the grid. Increasing the detection speed will affect detection performance inevitably since the detection task will be performed using less data. Therefore, in quickest detection algorithms there exists a trade-off between detection speed and detection reliability.

6.1 Bad Data Detection

6.1.1 Bad Data Definition

State estimation process aims to detect measurement errors, and suppress these errors if it is possible. Measurements may contain abnormally large errors that affect the accuracy of state estimation, called bad data, because of device failure, device misconfiguration, telecommunication medium, transient in power system, transient meter failure or malfunction, etc. On the other hand, bad data can be defined as a large abrupt change of short duration in observation window. Bus voltage data is illustrated as an example for bad data in Figure 6.1. Bad data can be clearly seen as a spike.

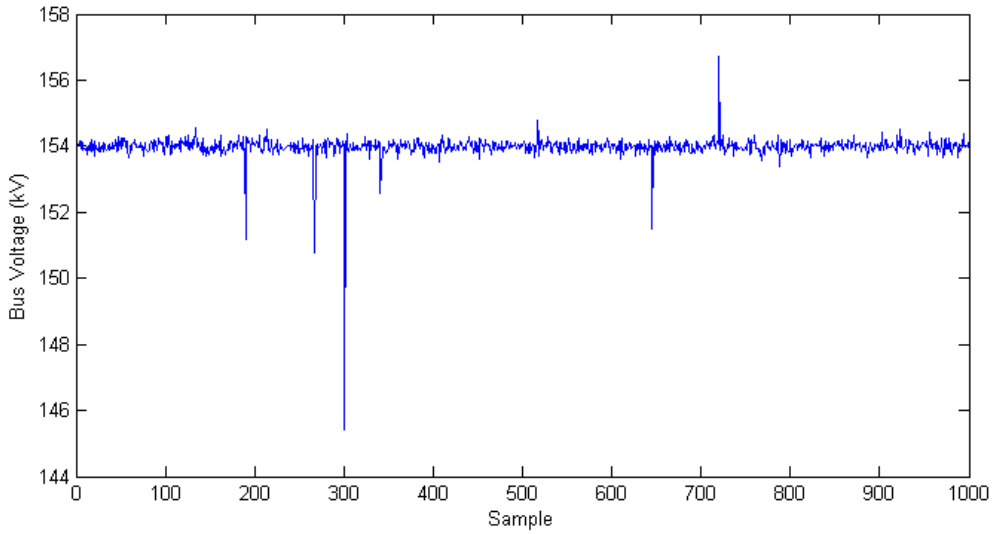

Figure 6.1 Bad data illustration.

Some kind of bad data are obvious and can be detected and eliminated by simple possibility checks. Negative voltage magnitudes, large differences between incoming and outgoing currents at a bus can be given as examples (Abur & Exposito, 2004).

6.1.2 Bad Data Detection Techniques

6.1.2.1 Chi-Square Distribution

Let X_i be a set of N independent random variables which have standard normal distribution, $X_i \sim N(0,1)$. A new random variable defined as:

$$Y = \sum_{i=1}^{N} {X_i}^2 \tag{6.1}$$

will have a Chi-square, X^2, distribution with N degrees of freedom, ${Y_i} \sim {X_N}^2$. N represents the number of independent variables in the sum. Let $\hat{f}$ be a function which is written in terms of measurement error (Abur & Exposito, 2004):

$$\hat{f} = \sum_{i=1}^{m} {R_{ii}}^{-1} {e_i}^2 = \sum_{i=1}^{m} \left(\frac{e_i}{\sqrt{R_{ii}}}\right)^2 = \sum_{i=1}^{m} ({e_i}^N)^2 \tag{6.2}$$

where R_{ii} is diagonal element of measurement covariance matrix, e_i is i th measurement error, m is the total number of measurements, and ${e_i}^N$ is normalized error. From statistical theory, $\hat{f}$ has chi-square distribution. Let n be the minimum number of measurements which satisfies the power balance equations in a power system. Hence, the maximum number of linearly independent measurement errors is $(m - n)$. This number also shows the degrees of freedom (Abur & Exposito, 2004).

An illustration of X^2 probability density function is shown in Figure 6.2. x_t is threshold value that represents the largest acceptable value for X which means no bad data is existed. The area which is located to the right of x_t represents probability of error. If the measured value of X exceeds x_t, the measured X will not have a X^2 distribution, i.e. bad data will be detected (Abur & Exposito, 2004).

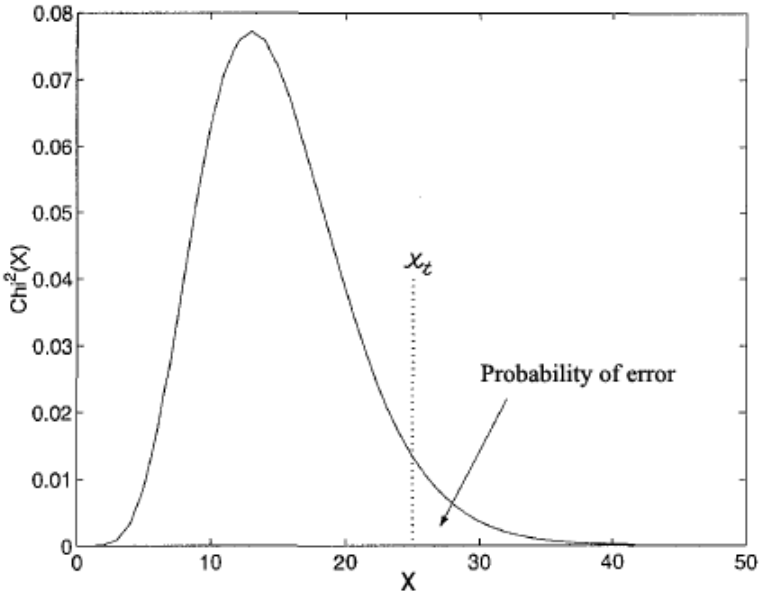

Figure 6.2 X^2 probability density function (Abur & Exposito, 2004).

6.1.2.2 Normalized Residuals

Due to the approximation of errors by residuals in Equation 6.2, Chi-square test may fail in certain cases. Normalized residuals test gives more accurate results than Chi-square test for detecting bad data (Abur & Exposito, 2004). Residual vector can be shown as follows:

$$r_i = z_i - h_i(\hat{x}) \tag{6.3}$$

where z_i represents measurement vector and $h_i(\hat{x})$ represents non-linear measurement function. Normalized residual can be calculated by simply dividing ith residual value by the corresponding diagonal entry in the residual covariance matrix.

$$r_i^N = \frac{|r_i|}{\sqrt{\Omega_{ii}}} \tag{6.4}$$

Presence of bad data can be detected by using statistical threshold test over r_i^N. If maximum r_i^N is equal to threshold or smaller than threshold, $\max_i r_i^N \leq \gamma$, normal hypothesis H_0 is accepted which means no bad data. Otherwise, alternative

hypothesis H_1 is accepted which means bad data is detected. Threshold value can be selected based on the desired sensitivity level.

6.2 False Data Injection Attack Detection

6.2.1 False Data Injection Model

The attacker may want to inject bad data to measurements for reaching its goal. For instance, attacker may try to directly compromise meters in power grid or attack RTUs for manipulating measurements which are collected by SCADA in the control center (Liu, Ning, & Reiter, 2011). Because the SCADA/EMS systems are connected to office LANs in control center, it is possible to attack these systems through internet connection (Sandberg, Teixeira, & Johansson, 2010).

Most of bad data detection and identification techniques in DC power flow model depend on the same assumption. If bad data occurs, the squares of differences between measurement and its corresponding estimate often become significant. This assumption is no longer valid, if the attacker knows the power system configuration. As a result of this situation, attacker may generate bad data without triggering the bad data detection alarm. This type of data is called false data injection attack (Liu, Ning, & Reiter, 2011).

The cyber data injection attack which is based on DC power flow model can be defined as (Liu, Ning, & Reiter, 2011),

$$z = Hx + a + e \tag{6.5}$$

where a is malicious data injected by attacker. The attack may be executed by one single attacker or by a group of coordinated attackers (Cui et al., 2012). It is assumed that H matrix is fully known by the control center. Attacker's knowledge about H is not known. If any injection attack occurs, it should be detected in a reliable way.

Hypothesis test can be used for defining situations in detection process. H_0 represents normal situation and H_1 represents attacked situation. If true hypothesis and decision are both H_1, it is named detection. If true hypothesis is H_1 and the

decision is H_0, it is called missed detection. If true hypothesis is H_0 and decision is H_1, it is called false alarm.

Table 6.1 Detection, misdetection, and false alarm terms

	True Hypothesis	**Decision**
Detection	H_1	H_1
Missed Detection	H_1	H_0
False Alarm	H_0	H_1

If attacker has knowledge about power system, attacker can mislead the control center by adding $a = Hc$ on the measurement vector. Therefore, the measurement vector is changed as follows (Liu, Ning, & Reiter, 2011).

$$z = H(x + c) + e \tag{6.6}$$

Therefore, the operator believes that true state is $(x + c)$, and c can be arbitrarily selected by attacker. Traditional statistical tests cannot detect these stealth attacks, because the attack vector stays in the range of H matrix (Cui et al., 2012).

When the attack is injected into the smart grid, it is obvious that mean of the measurement vector is shifted. Assume that normal state has Gaussian distribution $N(\mu_0, \sigma)$, or alpha-stable distribution $S(\alpha, \beta, \gamma, \mu_0)$. In both cases, initial mean value μ_0 turns to μ_1 when the attack comes out. The binary hypothesis can be expressed as below,

$$\begin{cases} H_0: \ z \sim N(0, \sigma^2) \\ H_1: \ z \sim N(a_t, \sigma^2) \end{cases} \tag{6.7}$$

where a_t is unknown attacker vector which is injected at random time τ. T_h represents the change detection time or stopping time. If $T_h < \tau$, the detector is alarmed before the change which is named false alarm. If $T_h > \tau$, detection delay is $T_h - \tau$. For quickest detection, the problem is minimizing the delay time. Page's CUSUM algorithm is an efficient tool for solving minimum delay problem. The

problem can be expressed as follows (Huang, Werner, Huang, Kashyap & Gupta, 2012).

$$T_d = sup_{\tau \geq 1} E_\tau[T_h - \tau \mid T_h > \tau] \quad (6.8)$$

It was discussed that smart grid must be protected from attacks for preventing problems. If attacker has knowledge about power system, it can mislead the control center through injecting false data and paralyze the power facility. The false data injection attack should be detected as quickly as possible for security of smart grid. The near real-time analysis to detect change of statistical behavior of state estimation is executed by control center for preventing possible future damage on whole network. Speed of the detection of any malicious attack has a vital importance to enable defence strategies in a moderate time in the grid. The delay between attacking time and detection time is desired to be as little as possible. This type of problem is called quickest detection problem.

Quickest detection algorithm tries to detect change as quickly as possible based on real time measurements when pre-defined conditions are met. Pre-defined conditions define the decision rules that optimize the trade-off between the detection speed and detection reliability (Huang, Werner, Huang, Kashyap & Gupta, 2012).

6.2.2 Two-Sided CUSUM Algorithm for Detecting False Data Injection Attack

Classification of quickest detection includes Bayesian and Non-Bayesian framework. Non-Bayesian framework is suitable for our approach, because of the prior probability of the attack and attacker vector are not known. In other words, a change of unknown distribution to unknown distribution is wanted to be detected at random time. Page's CUSUM algorithm is an efficient tool combining statistical hypothesis test for quickest detection problem (Huang, Werner, Huang, Kashyap & Gupta, 2012).

CUSUM is a sequential analysis technique for change detection which is developed by E. S. Page. Our detection approach depends on the two-sided CUSUM algorithm in tabular form for detecting increase or decrease in the mean of

measurements. Two-sided algorithm is easy to implement. We assumed that no attack is injected initially. After the attack is injected, the mean value of measurement is changed to $\mu_1{}^+ = \mu_0 + \delta\sigma$ or $\mu_1{}^- = \mu_0 - \delta\sigma$. The resulting alarm time can be shown as follows (Basseville & Nikiforov, 1993).

$$\begin{aligned} t_a &= \min\{\, i : (g_i{}^+ \geq h) \cup (g_i{}^- \geq h) \,\} \\ g_i{}^+ &= g_{i-1}{}^+ + z_i - \mu_0 - K \\ g_i{}^- &= g_{i-1}{}^- - z_i + \mu_0 - K \end{aligned} \tag{6.9}$$

z_i is the i th measurement value. If $g_i{}^+$ or $g_i{}^-$ exceeds the threshold h, it means that the injected false data is detected. K is one-half the magnitude of the shift. K is usually called the reference value or the allowance. But the problem is that how we should design the K and h parameters. In most practical cases, little knowledge about K parameter is existed. It is often chosen about halfway between μ_0 and μ_1. K parameter can be expressed in standard deviation units as below (Montgomery, 2000).

$$K = \frac{\delta\sigma}{2} = \frac{|\mu_1 - \mu_0|}{2} \tag{6.10}$$

δ is the amount of shift in the measurement mean that we wish to detect. The reasonable value for h is five times the process standard deviation τ (Montgomery, 2000). Beginning of the detection process, we assumed N initial measurement is not attacked. Hence, σ can be estimated from initial measurements and it is updated for every predefined interval value before the processing point. Consequently, threshold value is updated online.

The tabular form of two-sided CUSUM works by accumulating derivations from μ_0 that are above target with first statistics $S_i{}^+$ and accumulating derivations from μ_0 that are below target with second statistics $S_i{}^-$ (Montgomery, 2000). The initial values set $S_i{}^+ = S_i{}^- = 0$. The tabular form of CUSUM can be expressed as follows.

$$\begin{aligned} S_i{}^+ &= \max(0\,,\, S_{i-1}{}^+ + z_i - \mu_0 - K) \\ S_i{}^- &= \max(0\,,\, S_{i-1}{}^- - z_i + \mu_0 - K) \end{aligned} \tag{6.11}$$

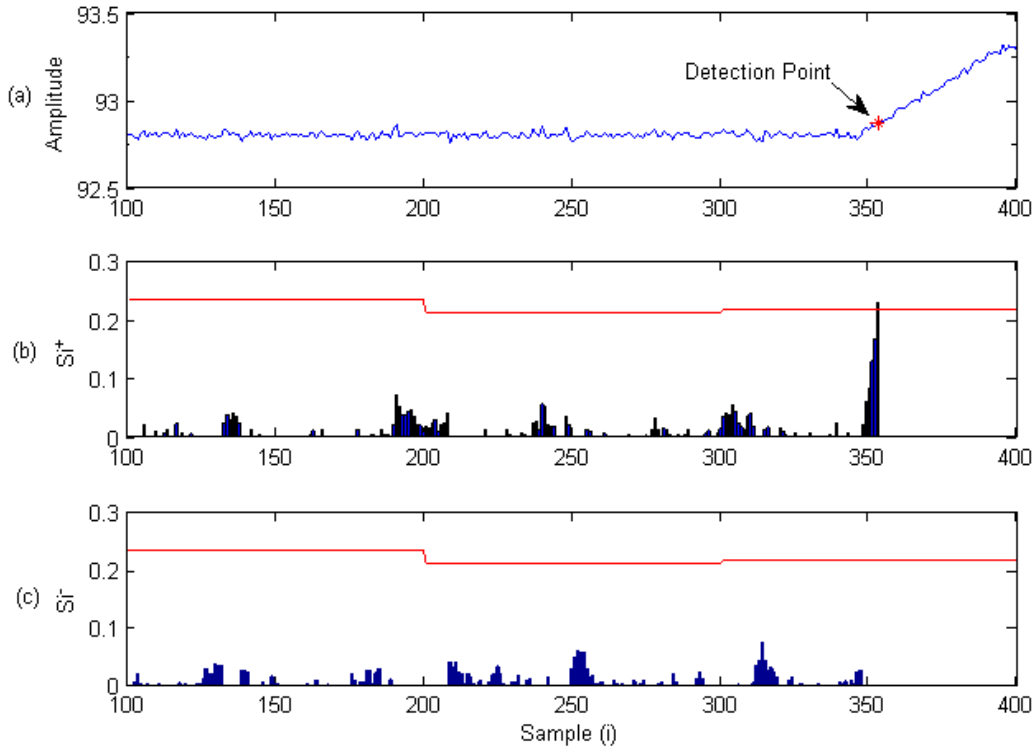

Figure 6.3 (a)An example signal. (b) S_i^+ is upper CUSUM. (c) S_i^- is lower CUSUM.

An example is illustrated in Figure 6.3 for showing how tabular form of CUSUM works on attacks. We assumed that there is no attack, initially. The attack is randomly generated at unknown time. After the attack starting point, the data is increased with 0.01 step size until it reaches half a unit DC offset. In Figure 6.3 (a), it is seen that data trend is changed. On the other hand, attacker started to inject false data attack. In Figure 6.3 (b), red horizontal line represents the threshold which is updated in every 100 sample for this example. Threshold update process depends on variance of the last 100 sample. Data attack is detected when S_i^+ exceeds the threshold. The CUSUM algorithm is terminated after threshold is exceeded. In Figure 6.3 (c), S_i^- statistic is shown. If data trend tends to lower side, S_i^- may exceed the threshold and stop the detecting process.

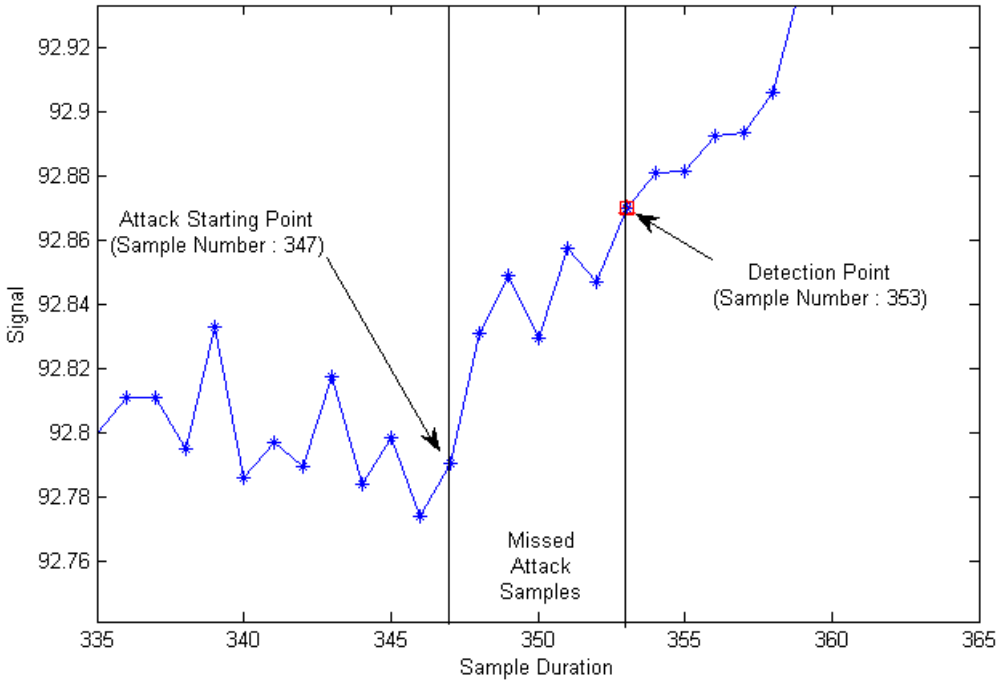

Figure 6.4 Illustration of attacking point, missed attack samples, and detection point.

Figure 6.4 shows the attacking point, missed samples and detection point that CUSUM approach determined. τ is the unknown attacking time and T_h is the stopping time. The delay or average run length (ARL) is $T_h - \tau$. ARL is 6 and number of misdetection is 5 for this example. ARL shows the performance of detection speed. Quickest detection aims to make decision with minimum ARL. But, in quickest detection algorithms there exists a trade-off between detection speed and detection reliability. Because of this reason, the threshold value should be selected as optimum for operation sensitivity. The higher threshold selection increases the decision time and detection reliability. The smaller threshold selection decreases decision time but it may increase the number of false alarms (Huang, Werner, Huang, Kashyap, & Gupta, 2012).

6.3 A False Data Injection Attack Scenario and Its Detection

In previous section, we investigated CUSUM algorithm for the detection of attacks. Now, we made up a scenario to show the possible idea that underlies false data injection attack. Node quickest detection is implemented and we used IEEE-14

bus topology as shown in Figure 6.5 for our scenario. Data is generated by using MATPOWER in MATLAB. It is assumed that power grid is stable before the attack and no failure is existed in power grid during experiment.

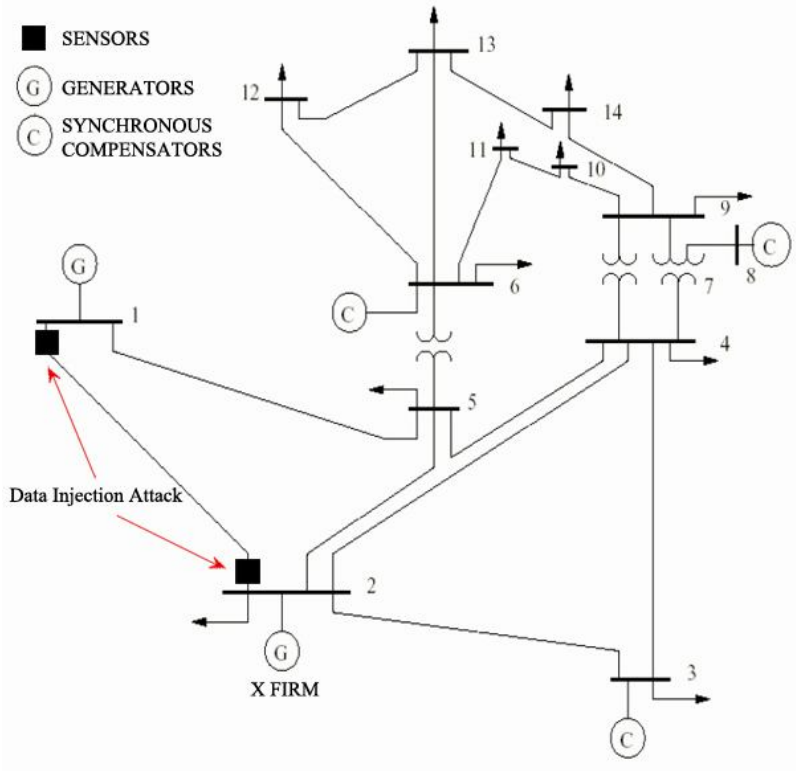

Figure 6.5 IEEE-14 Bus System.

Attacker wants to manipulate the data that is sent by sensors, which are shown above, for misleading system operator. Attacker obtained some unauthorized information and also knew the power line is under stress. We assumed that the power line between Bus-1 and Bus-2 is nearly full of capacity. Power flow direction is from Bus-1 to Bus-2. If the load of this branch is increased, the over-current relay may send “open” signal to the circuit-breakers which are located at the beginning of the line and at the end of the line. This situation may trigger other failures in power grid and may cause power blackout. Hence, attacker wants to show this information is wrong and injects wrong data to communication channel. X-FIRM has a power plant which is connected to Bus-2. If attacker injects the false data to overestimate this line load, system operator may send “produce more power” signal to X-FIRM for decreasing the load of this branch. Consequently, X-FIRM earns much money with

injecting bad data. Power flow data which is related to our experiment is shown below for IEEE-14 bus system. This data represents mean values of power flow measurements for certain interval of time.

Table 6.2 Power flow data.

			From Bus Injection		To Bus Injection	
Branch No	From Bus	To Bus	P(MW)	Q(MVAr)	P(MW)	Q(MVAr)
1	1	2	**156,88**	-20,40	**-152,59**	27,68
2	1	5	75,51	3,85	-72,75	2,23
3	2	3	73,24	3,56	-70,91	1,60
4	2	4	56,13	-1,55	-54,45	3,02
5	2	5	41,52	1,17	-40,61	-2,10
6	3	4	-23,29	4,47	23,66	-4,84
7	4	5	-61,16	15,82	61,67	-14,20
8	4	7	28,07	-9,68	-28,07	11,38
9	4	9	16,08	-0,43	-16,08	1,73
10	5	6	44,09	12,47	-44,09	-8,05
11	6	11	7,35	3,56	-7,30	-3,44
12	6	12	7,79	2,50	-7,71	-2,35
13	6	13	17,75	7,22	-17,54	-6,80
14	7	8	0,00	-17,16	0,00	17,62
15	7	9	28,07	5,78	-28,07	-4,98
16	9	10	5,23	4,22	-5,21	-4,18
17	9	14	9,43	3,61	-9,31	-3,36
18	10	11	-3,79	-1,62	3,80	1,64
19	12	13	1,61	0,75	-1,61	-0,75
20	13	14	5,64	1,75	-5,59	-1,64

Attacker wants to change the power flow data which is measured from power line between Bus-1 and Bus-2, and make the system operator believe this line is nearly full of capacity. The data which is attacked by attacker is marked on the table. Attack increasing step for every sample is 0.01 until it reaches 5 units DC shift. Attacks are individually performed at unknown time for 10000 times. The noise has alpha-stable distributions, $S\,(1.8, 0, 0.01, 0)$. The CUSUM threshold value is increased for illustrating the impact of threshold on detection time, detection ratio, and false alarm ratio.

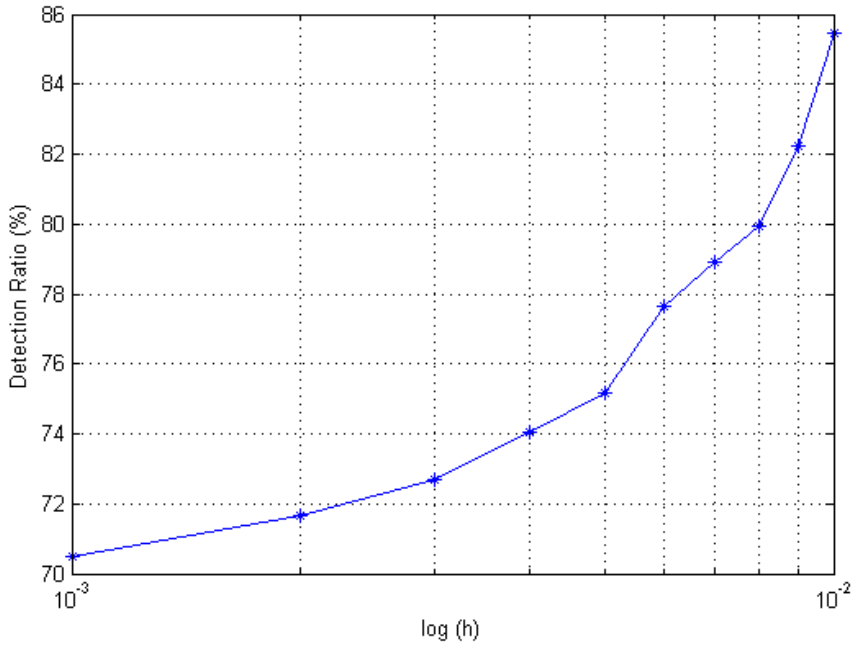

Figure 6.6 Detection ratio (%) versus threshold.

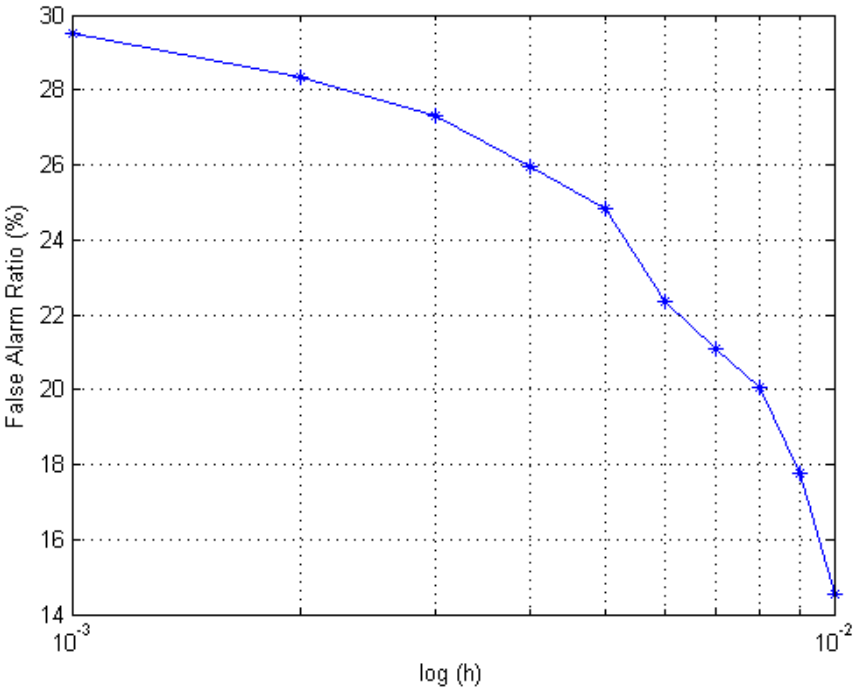

Figure 6.7 False alarm ratio (%) versus threshold.

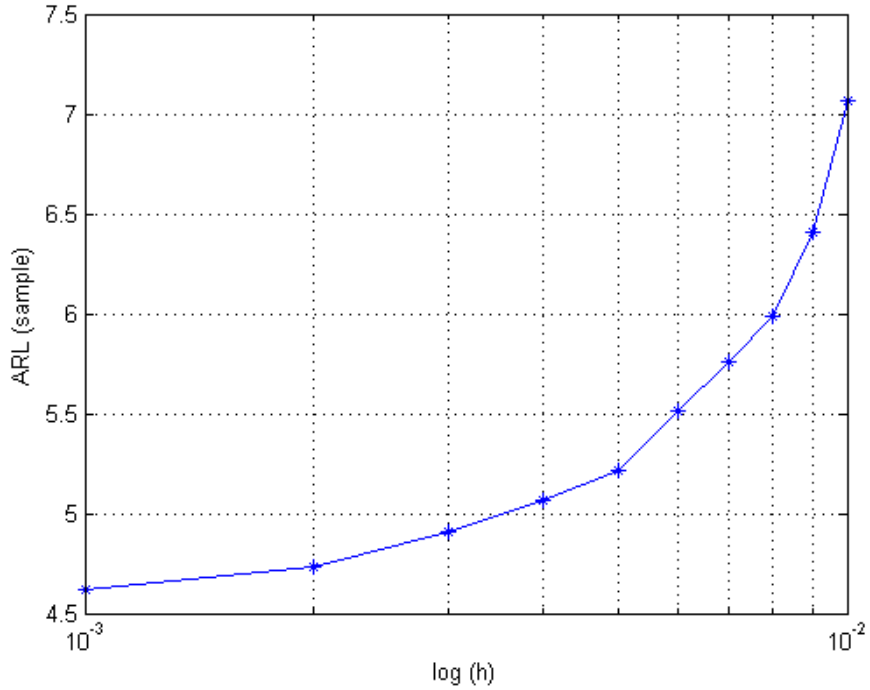

Figure 6.8 ARL versus threshold.

In Figure 6.6, it's clearly seen that if the threshold value is increased, the detection ratio increased. Detection ratio is calculated by dividing the number of detected attacks to realization number. In Figure 6.7, if we increase threshold value, number of false alarms will statistically decrease which corresponds to improved reliability. Figure 6.8 shows that threshold and ARL is both increasing. On the other hand, CUSUM needs more time for making a decision when threshold rises.

6.4 Performance Tests of CUSUM for Alpha-Stable Distributions

Performance tests are implemented for illustrating behavior of CUSUM algorithm in impulsive noise environment for different alpha values. Most of studies assume that noise has Gaussian distributions. However, processes in practice are generally impulsive in nature and are not well described with Gaussian distribution. Hence, the noise is modeled with alpha-stable distributions which is well-suited for describing impulsive components. As mentioned in Chapter Five, if α decreases, the existence rate and the strength of the outliers increase. It is expected that if the impulsiveness increases, the performance of CUSUM for detecting attacks will decrease. Alpha

value is selected from 1.5 to 1.9. 10000 individual runs are performed for each alpha value. Data length is 1000 sample during the test.

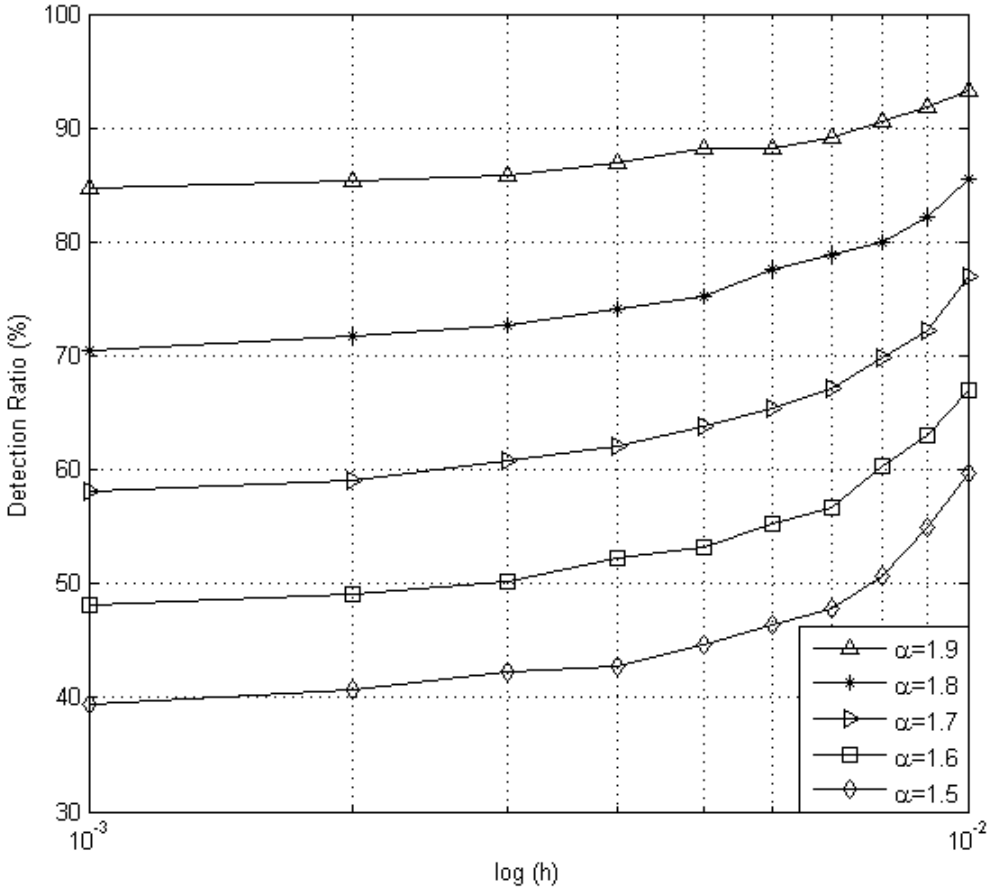

Figure 6.9 Detection ratio (%) versus threshold.

As expected, if alpha decreases, impulsive components and their occurrences will increase. Therefore, detection ratio dramatically decreases as seen in Figure 6.9. Because of the strong impulses which occurred before the attack exceed the predefined threshold, CUSUM algorithm is terminated before the attack in impulsive environments. The threshold can be selected greater for improving performance of detection in impulsive environments.

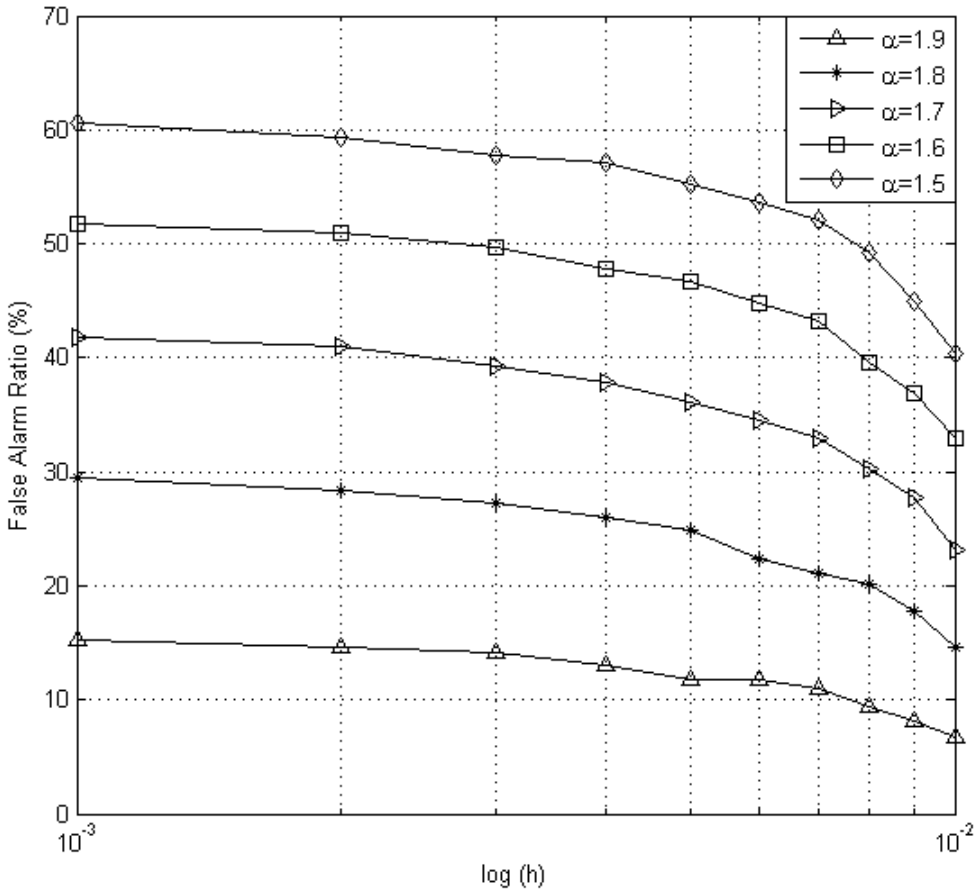

Figure 6.10 False alarm ratio (%) versus threshold.

In Figure 6.10, it is seen that false alarm ratio increases when alpha value decreases. In other words, the false data detector cannot catch the attacks because of the impulses which existed before attack. Reliability of detection can be improved by increasing the CUSUM threshold. Higher threshold contributes decreasing number of false alarms but increases the decision time.

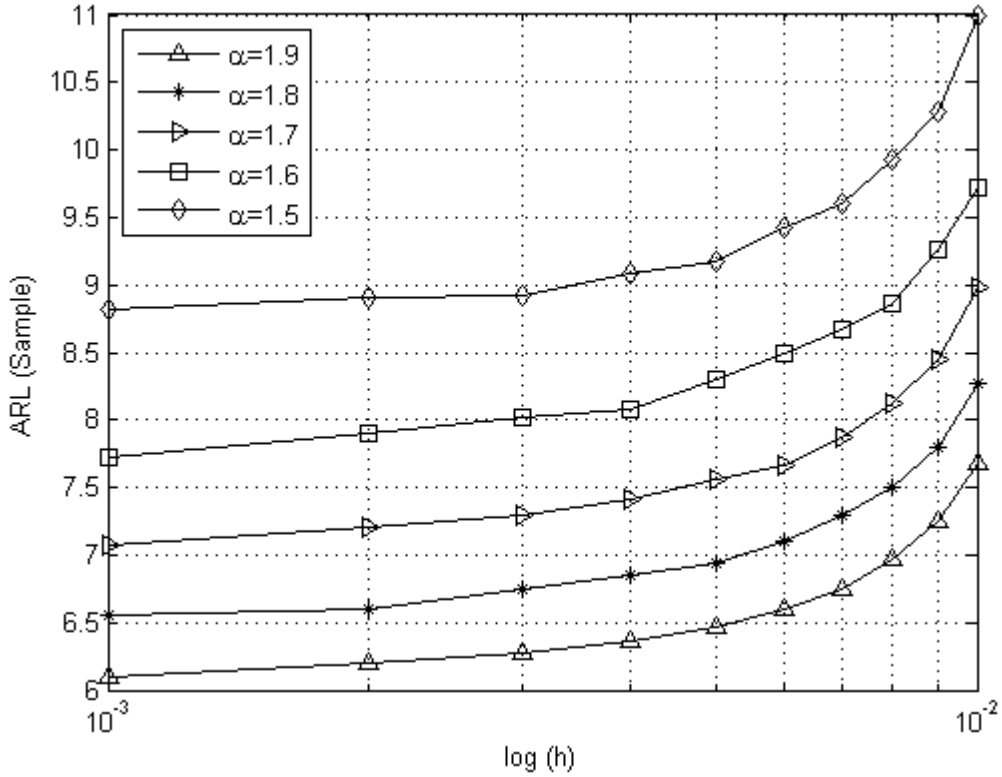

Figure 6.11 ARL versus threshold.

In Figure 6.11, higher alpha values provide less average run length. CUSUM needs more time for making a decision in very impulsive environments. It is known that the detection ratio and ARL are related. Threshold value can be selected higher in very impulsive environments for increasing detection ratio, but this makes decision time longer.

CHAPTER SEVEN
CONCLUSIONS

One of the main contributions in this thesis is that when the noise becomes more impulsive in state estimation problem of the smart grid, the myriad filter has the best performance for the low signal to dispersion ratios. Similarly, the sample meridian filter follows the performance of the myriad filter and can also be preferable for low signal to dispersion ratios.

Interestingly, weighted least squares filter is preferable compared with myriad, meridian, and median filters for high signal to dispersion ratios and characteristic exponents near 2. The monotonocially decreasing error performance is observed from the sample median filter and can be preferable for high SDRs and relatively smaller characteristic exponents.

Two-sided CUSUM in tabular form is easy to implement for detection of attacks. However, threshold value selection is crucial for performance of detection. Higher threshold value selection makes the detection reliable, also decreases number of false alarms, but increases average run length. There is a trade-off between detection ratio and average run length. The optimal threshold value should be selected according to operation sensitivity.

According to simulation results, if characteristic exponent value decreases, performance of detection will dramatically decrease and number of false alarms will extremely increase. In very impulsive environments, the threshold value should be selected as high as possible for improving detection reliability and preventing false alarms.

REFERENCES

Abur, A., & Exposito, A. G. (2004). *Power system state estimation: Theory and implementation.* New York: Marcel Dekker.

Arce, G. R. (2005). *Nonlinear signal processing: A statistical approach*. New Jersey: John Wiley & Sons.

Aysal, T. C., & Barner, K. E. (2007). Meridian filtering for robust signal processing. *Signal Processing, IEEE Transactions on*, *55*(8), 3949-3962.

Basseville, M., & Nikiforov I. V. (1993). *Detection of abrupt changes: Theory and application.* New Jersey: Prentice Hall.

Bi, S., & Zhang, Y. J. A. (2011). Defending mechanisms against false-data injection attacks in the power system state estimation. *GLOBECOM Workshops, 2011 IEEE*, 1162-1167.

Bobba, R. B., Rogers, K. M., Wang, Q., Khurana, H., Nahrstedt, K., Overby, T. J. (2010). Detecting false data injection attacks on DC state estimation. *Preprints of the First Workshop on Secure Control Systems, CPSWEEK,* 2010.

Cui, S., Han, Z., Kar, S., Kim, T.T., Poor, H. V., Tajer, A. (2012). Coordinated data-injection attack and detection in smart grid. *Signal Processing Magazine, IEEE, 27*(5), 106-115

Gonzales, J. G., Griffith, D.W., & Arce, G. R. (1996). Matched myriad filtering for robust communications. *Proceeding of the 1996 CISS*, 1996.

Huang, Y. F., Werner, S., Huang, J., Kashyap, N. & Gupta, V. (2012). State estimation in electric power grids: Meeting new challenges presented by the requirements of the future grid. *Signal Processing Magazine, IEEE*, *29*(5), 33-43.

Huang, Y., Li, H., Campbell, K. A. & Han, Z. (2011). Defending false data injection attack on smart grid network using adaptive cusum test. *Information Sciences and Systems (CISS), 2011 45th Annual Conference on,* 1-6. Retrieved March 23, 2013, from IEEE.

Kay, S. M. (1998). *Fundamentals of statistical signal processing, volume II: Detection theory.* New Jersey: Prentice Hall.

Korres, G. N., & Manousakis, N. M. (2011). State estimation and bad data processing for systems including PMU and SCADA measurements. *Electric Power Systems Research*, *81*(7), 1514-1524.

Kosut O., Liyan J., Thomas R. J. & Lang T. (2010). Malicious data attacks on the smart grids, *IEEE Transactions on Smart Grids*, *2*(4), 645-658.

Liu, Y., Ning, P. & Reiter, M. K. (2011). False data injection attacks against state estimation in electric power grids. *ACM Transactions on Information and System Security (TISSEC)*, *14*(1), 13.

MíDnguez, R., Conejo, A. J. & Hadi, A. S. (2008). Non Gaussian state estimation in power systems. In *Advances in Mathematical and Statistical Modeling*, 141-156. Retrieved April 16, 2013, from Birkhäuser Boston.

Montgomery, D. C. (2000). *Introduction to statistical quality control* (4th ed.). New York: Wiley.

Monticelli, A. (2000). Electric power system state estimation. *Proceedings of the IEEE*, *88*(2), 262-282.

Pander, T., & Przybyła, T. (2012). Impulsive noise cancelation with simplified Cauchy-based p-norm filter. *Signal Processing*, *92*(9), 2187-2198.

Rahman, M., & Mohsenian-Rad, H. (2012). False data injection attacks with incomplete information against smart power grids. *Global Communications Conference (GLOBECOM), 2012 IEEE*, 3153-3158. Retrieved January 23, 2013, from IEEE database.

Samoradnitsky, G., & Taqqu, M. S. (1994). *Stable non-Gaussian random processes: Stochastic models with infinite variance*. Florida: CRC Press.

Sandberg, H., Teixeira, A. & Johansson, K. H. (2010). On security indices for state estimators in power networks. *Preprints of the First Workshop on Secure Control Systems, CPSWEEK 2010, Stockholm, Sweden*. 2010, 1–6.

Veillette, M. (2012). *Alpha-stable distributions in MATLAB.* Retrieved March 5, 2013, from http://math.bu.edu/people/mveillet/html/alphastablepub.html.

Wu, F. F., Moslehi, K. & Bose, A. (2005). Power system control centers: Past, present, and future. *Proceedings of the IEEE*, *93*(11), 1890-1908.

Zimmerman, R. D., Murillo-Sánchez, C. E. & Gan, D. (2011). *MATPOWER, a MATLAB power system simulation package.* Retrieved February 25, 2013, from http://www.pserc.cornell.edu/matpower/.

APPENDICES

```
function
[v,va,myra,myrph,mera,merph,meda,wlsa,wlsph,SDR]=genfilt(alpha)

%genfilt.m
%This program generates isotropic complex alpha-stable test data
%and filters generated noisy data for myriad,meridian,median and wls
filters.

%Bus System Properties
voltage=1.045;
phang=-12.72;
re=voltage*cos(phang);
im=voltage*sin(phang);
vol=re+1i*im;

gamma=cos(pi*(alpha/4))^(2/alpha);
beta=1;
delta=0;

sample=1000;
realization=200;

%Generate dispersion for desired dB
dbstart=0; dbstep=3; dbfinish=30; say=(dbfinish-(dbstart))/dbstep;
dispersion=zeros(1,say); k=1; SDR=zeros(1,say);

for db=dbstart:dbstep:dbfinish
    SDR(1,k)=db;
    tempv=db/20;
    tempv=10.^(tempv);
    tempv=1/( sqrt(2)*(tempv/voltage) );
    dispersion(1,k)=tempv^alpha;
    k=k+1;
end

SDR=sort(SDR,'descend');
dispersion=sort(dispersion);

        noise=zeros((say*realization),sample);
        v=zeros((say*realization),sample);
        vr=zeros((say*realization),sample);
        vi=zeros((say*realization),sample);
        va=zeros((say*realization),sample);


b=1;
    for i=1:1:(say+1)

      for j=1:realization

      %isotropic complex alpha-stable noise
      G1=stblrnd(2,0,gamma2,0,1,sample);
```

```
        G2=stblrnd(2,0,gamma2,0,1,sample);
        if alpha==2
        G=G1+1i*G2;
        noise(b,:)=G;
        else
        A = stblrnd(alpha/2,beta,gamma,delta,1,sample);
        X=(A.^(1/2)).*(G1+1i*G2);
        noise(b,:)=X;
        end

        v(b,:)=noise(b,:)+vol; %noisy complex voltage

        vr(b,:)=real(v(b,:));
        vi(b,:)=imag(v(b,:));
        va(b,:)=abs(v(b,:));

        %Myriad
        myrr(b,:)=smyriad2(vr(b,:),21,alpha,gamma2);
        myri(b,:)=smyriad2(vi(b,:),21,alpha,gamma2);
        [myrph(b,:),myra(b,:)]=cart2pol(myrr(b,:),myri(b,:));

        %Meridian
        merr(b,:)=smeridian2(vr(b,:),21,alpha,gamma2);
        meri(b,:)=smeridian2(vi(b,:),21,alpha,gamma2);
        [merph(b,:),mera(b,:)]=cart2pol(merr(b,:),meri(b,:));

        %WLS
        wlsr(b,:)=wlsfilt(vr(b,:));
        wlsi(b,:)=wlsfilt(vi(b,:));
        [wlsph(b,:),wlsa(b,:)]=cart2pol(wlsr(b,:),wlsi(b,:));

        %Median
        meda(b,:)=medfilt1(va(b,:),20);

        b=b+1;
        end

    end
```

```
function
[mede,mere,myre,wlse]=errcalc(Vact,SDR,meddata,merdata,myrdata,wlsda
ta,start,endd,alpha)

%errcalc.m
%This program calculates FLOE and plots FLOE vs. SDR for filtered
%data.

n=endd-start;
k=start-1;
t=length(SDR);
realization=200;

mede=zeros(1,t);
mere=zeros(1,t);
```

```
myre=zeros(1,t);
wlse=zeros(1,t);

alpha=alpha-0.001;
s=1; %line number

%FLOE
for x=1:t
    for j=1:realization

        for p=1:n
            mede(1,x)=mede(1,x)+(abs(Vact(1,p+k)-
meddata(s,p+k)).^alpha);
            mere(1,x)=mere(1,x)+(abs(Vact(1,p+k)-
merdata(s,p+k)).^alpha);
            myre(1,x)=myre(1,x)+(abs(Vact(1,p+k)-
myrdata(s,p+k)).^alpha);
            wlse(1,x)=wlse(1,x)+(abs(Vact(1,p+k)-
wlsdata(s,p+k)).^alpha);
        end
        s=s+1;
    end
            mede(1,x)=(1/n)*mede(1,x)*(1/realization);
            mere(1,x)=(1/n)*mere(1,x)*(1/realization);
            myre(1,x)=(1/n)*myre(1,x)*(1/realization);
            wlse(1,x)=(1/n)*wlse(1,x)*(1/realization);
end

figure;
semilogy(SDR,smooth(myre),'r-s'); grid; hold on;
semilogy(SDR,smooth(mere),'k-^');
semilogy(SDR,smooth(mede),'m-*');
semilogy(SDR,smooth(wlse),'b-o');

tit=sprintf(' = %1.1f',alpha);
s=strcat('\alpha',tit);
title(s);
legend('Sample Myriad',...
    'Sample Meridian',...
    'Sample Median',...
    'WLS');
xlabel('SDR (Signal to Dispersion Ratio)');
ylabel('FLOE (Fractional Lower Order Error)');
```

```
function [output]=smyriad(data,N,alpha,gamma)

%smyriad.m
%This program calls myriad filter (myriad.m) file in recursive way.


%initialization
size=length(data);
halfwin=(N-1)/2;
range=size-halfwin;
output=zeros(1,range);
```

```
%Linearity Parameter Formula
best_k=(gamma.^(1/alpha))*tan(pi*alpha*0.25);

%Myriad Filtering Loop
    for p=(halfwin+1):range
        output(1,p)=myriad(data(1,(p-halfwin):(p+halfwin)),best_k);
    end;

output(~output)=nan; %skip zeros
```

```
function minbeta = myriad(samplewin,k)

%myriad.m
%This program filters data using myriad method.
%k is linearity parameter.

xmin = min(samplewin);
xmax = max(samplewin);
betamin = xmin;

N=length(samplewin);
trans = zeros(1,N);

trans = samplewin - xmin;
trans = trans.^2;
trans = trans + k.^2;
trans=abs(trans);
trans=log(trans);
minimum = prod(trans);

for range = xmin:0.01:xmax
    trans = samplewin - range;
    trans = trans.^2;
    trans = trans + k.^2;
    trans=abs(trans);
    trans=log(trans);
    cumulative = prod(trans);

        if (cumulative < minimum)
            betamin = range;
            minimum = cumulative;
        end;
end;

minbeta = betamin;
```

```
function [output]=smeridian(data,N,alpha,gamma)

%smeridian.m
%This program calls meridian filter (meridian.m) file in recursive
way.
```

```
%initialization
size=length(data);
halfwin=(N-1)/2;
range=size-halfwin;
output=zeros(1,range);

%Medianity Parameter Formula from
best_k=(gamma.^(1/alpha))*tan(pi*alpha*0.25);

%Myriad Filtering Loop
    for p=(halfwin+1):range
        output(1,p)=meridian(data(1,(p-
halfwin):(p+halfwin)),best_k);
    end;

output(~output)=nan; %skip zeros
```

```
function minbeta = meridian(samplewin,k)

%meridian.m
%This program filters data using meridian method.
%k is medianity parameter.

xmin = min(samplewin);
xmax = max(samplewin);
betamin = xmin;

N=length(samplewin);
trans = zeros(1,N);

trans = samplewin - xmin;
trans = trans + k;
trans=abs(trans);
trans=log(trans);
minimum = prod(trans);

for range = xmin:0.01:xmax
    trans = samplewin - range;
    trans = trans + k;
    trans=abs(trans);
    trans=log(trans);
    cumulative = prod(trans);

        if (cumulative < minimum)
            betamin = range;
            minimum = cumulative;
        end;
end;

minbeta = betamin;
```

```
function [r,output]=randattack(x)

%randattack.m
%Random DC Attack Generator (with Non-Gaussian Noise)

step=0.01;    %Increasing step
size=200;    %Attack sample size
shift=0.5;     %DC shift magnitude
start=100;    %Attack after starting point

%noise properties
alpha=1.5;
dispersion=0.01;
beta=0;
delta=0;

datalength=length(x);
r =randi([start datalength],1,1);  %Define attacking time randomly.
attack=zeros(1,datalength);

%Construct Attack Vector
isize=shift/step;
horsize=size-2*isize;

%increasing
for i=1:1:isize
    attack(1,r+i)=i*step;
end

%horizontal attack data
attack(1,(r+isize):(r+isize+1+horsize))=shift;

%decreasing
for i=0:1:isize
    attack(1,r+isize+2+horsize+i)=shift-i*step;
end
%End of construction

attack(1,1:(r-1))=0;         %Make zero the data before attack.
attack(1,(r+size+1):datalength)=0;    %Make zero the data after
attack.

e=stblrnd(alpha,beta,dispersion,delta,1,datalength);
%e=randn(1,datasize);

output= x + attack(1,1:datalength);
output=output+e;
r=r+1;
```

```
function [shi,slo,mu,h,sd,detpoint]=fdd(x,far)

%Two-sided CUSUM chart in tabular form
```

```
%parameters
far=0.01; %False Alarm Rate
mdr=0.01; %Miss Detection Rate

delta=1; %the amount of shift in the process mean that we wish to
detect
N=100; %length of initally non-attacked data
mu=mean(x(1,1:N)); %mean of first samples
sd=std(x(1,1:N)); %variance of initial samples
k= delta * sd * 0.5; %the rise in the arm corresponding to one
sampling unit
d=(2/delta.^2)*log((1-mdr)/far);
hconst=3; a=1;
h(1,a)= d*k*hconst; %initial threshold

dl=length(x);
shi=zeros(1,dl);
slo=zeros(1,dl);

plot(x,'b'); hold on;
for i= N:1:dl

if(mod(i+1,N)==0)
    sd=std(x(1,i-N:i));
    k= delta * sd * 0.5;
    a=a+1;
    h(1,a)= d*k*hconst; %new threshold
end

shi(1,i)=max(0,shi(1,i-1)+x(1,i)-mu-k);
slo(1,i)=max(0,slo(1,i-1)-x(1,i)+mu-k);

if (shi(1,i)>h(1,a) || slo(1,i)>h(1,a))
    break; %something detected
end
end
    detpoint=i;
    plot(detpoint,x(1,detpoint),'r-^');
```